

Préparé par Dr. Neila Ben Lakhal , (PhD@Titech JAPAN)
E-mail : neila.benlakhal@gmail.com

Cours à assurer



- 2 cours :
  - Technologies Web (2ING)
  - SOA/ISSOA(3ING SI/Rx)





Selon vous, une Application Web c’est quoi ?

 neila.benlakhal@gmail.com

Ma Définition



- C'est une application qui a pour particularité :
  - Accessible via un navigateur (IE, Chrome, Safari, Firefox, etc.)
  - Elle s'adresse à une population hétérogène :
    - Des simples utilisateur(humains)
    - D'autres applications (Mobile / Desktop)
- Son contenu est hétérogène
  - Information structurée dans des SGBDs
  - Information non structurés (hypermédia)
- Fortement basée sur une structure navigationnelle (Hyperlinks) vertical ou horizontal
- Elle a pour but d'indexer, publier et maintenir de larges quantité de données.

neila.benlakhal@gmail.com

Exemples d'applications Web



- Exemples :
  - Orientée E-business, E-commerce
    - Catalogues online (Amazon, …)
    - Auction, vente aux enchères (Ebay, …)
  - Orientée service:
    - Online order tracking, Réservation en ligne (trip Advisor, Expedia,..)
  - Orientée communauté :
    - Portails web (yahoo, …)
    - Social networks (Facebook, Flickr,Google+…)
    - Forum,…
  - Orientée contenu :
    - Digital librairies, …
    - E-learning platforms (UVT)

Les technologies du Web



- Elles englobent les technologies de développement d'applications Web qui sont:
  - XHTML, CSS, Javascript, DOM, JQuery, Php, Perl, JSP, Ruby, ASP.NET, XML, Xquery, Ajax, JSON, Etc…

- Ainsi que les divers Frameworks et outils de gestion de contenu (CMS) tels que :
  - IBM WebSphere Portal,
  - ZendFramework
  - CodeIgniter,
  - phpNuke,
  - Symfony,
  - Ruby on Rail,
  - Etc..



- Avant de faire le tour de ces technologies,
  - À travers les cours,
  - Les Tps à rendre,
  - Les projets à élaborer,

- Quelques généralités sur le développement en environnement WEB.

Web ≠ Internet



- Le **Web** est un système de fichiers présent sur des machines (serveurs) transitant par un protocole particulier **HTTP**, consultable grâce à des navigateurs web et fonctionnant **SUR** Internet !



- **Internet** est un assemblage de multiples réseaux, tous connectés entre eux en utilisant le protocole **TCP/IP**. Cet amas de câbles, de fibres optiques... de matériels, pour faire simple, constitue Internet, aussi appelé « **le réseau des réseaux** ».
- **Internet est hardware; le Web est software**
  - *Il y a plusieurs autres applications basées sur Internet:*
    - *e.g., email, telnet, ftp, usenet, Instant Messenger, etc.*
    - *Elles utilisent des ports différents Web(80), e-mail(25) etc.*

URL, IP, DNS



- Tout périphérique est accessible sur un réseau par son adresse 74.125.19.147
- Lorsque vous demandez une page web à votre navigateur, vos tapez une adresse **URL** http://www.google.fr

http://www.monsite.com/dossier/fichier.html

Protocole de communication

Adresse du serveur

Arborescence sur le serveur

- Les serveurs **DNS** font le lien entre l'URL tapée et son adresse IP

Internet : Historique



- 1967 : Arpanet (réseau militaire américain robuste aux pannes)
- Les années 70s: ARPANET doublait de taille chaque année.
- 1984: ~1000 d'ordinateurs militaires et académiques connectés.
- 1992:
  - ~1,000,000 d'ordinateurs connectés.
  - Internet society a été crée et le contrôle lui a été transféré :
    - Internet Engineering Task Force (IETF).
    - Internet Architecture Board .
    - Internet Assigned Number Authority .
    - **World-Wide-Web Consortium (W3C).**

Internet : le nombre d'utilisateurs



- 2,279,709,629 Internet users in 2010 (approx. 28.7 % of the world's population)
- (http://www.internetworldstats.com/top20.htm)

(18 September, 2012)

- Japan: 101,228,736 users (approx. 80% of the population)
- Tunisia: 3,856,984 users (approx. 36 % of the population)

neila.benlakhal@gmail.com

Historique (bref) Web



- L'idée des liens hypertexte: a été proposée au début dans les années 40 par Vannevar Bush.
- En 1989: Tim Berners-Lee du European Particle Physics Laboratory (CERN) a conçu un système d'hypertexte pour connecter des document sur le Net.



Tim Berners-Lee

- Il l a conçu un langage pour spécifier le contenu des documents. Devenu par la suite : HyperText Markup Language (HTML).
- Il a conçu un protocole pour télécharger et interpréter le contenu des documents devenu par la suite HTTP.
- Il a implémenté le premier navigateur Mozaïc : seulement du texte, PAS d'objet multimédia.

Pour accéder au Web

- Pour accéder au Web (client),
- Il vous suffit d'avoir : un **navigateur**
  - Pour ce cours sur lequel la majorité des exemples ont été testés, les navigateur **Chrome 21** est conseillé

- Pour programmer du Web ?
- Développement Web (web application) vs. Développement application monoposte (Standalone application) ???



- Pour programmer, il vous faut **un serveur HTTP**:
  - **Microsoft IIS** (Internet Information Services)
  - Le serveur **Apache** (apache Foundation) (http://www.apache.org/)
  - Les environnements de développement:
    - (open-source) WAMP, LAMP, XAMPP permettent de simuler un environnement de développement Web.
    - (MICROSOFT) Microsoft Web Platform 4.0 : IIS+SQL SERVER 2008+ Visual studio express+.NET FRameworK.

    http://www.microsoft.com/web/downloads/platform.aspx

neila.benlakhal@gmail.com

Notion client/serveur



- Quand un visiteur veut aller sur un site Web. Il tape l'adresse URL, ok, mais ensuite ? La page s'affiche, d'accord, mais entre-temps que s'est-il passé ?
  - Lorsque l'internaute tape l'adresse d'un site dans son browser (**client**) (www.google.fr par exemple), celui-ci envoie une requête au **serveur** qui héberge ce site. Le serveur transmet alors la page demandée au browser qui l'affiche.
  - Le **client**, c'est le navigateur Internet de l'internaute comme, Mozilla Firefox, Microsoft IE, Apple Safari, Google Chrome, Opera etc.

Static vs. Dynamic pages



- « A static web page is a web page that always comprises the same information in response to all download requests from all users. »
  - Le contenu (texte, multimedia, etc.) est toujours le même.
- Avantages
  - Rapide et facile à mettre en place même pour les non connaisseurs.
  - Un outil idéal pour donner un avant gout/maquette d'un site Web à construire.
- Inconvénients
  - N'offrir qu'une et une seule présentation, sans aucune possibilité de personnalisation et avec une interactivité limitée au strict minimum
  - La maintenance des site Web de taille est couteuse.
  - Difficile de les garder à jour et surtout cohérent.

neila.benlakhal@gmail.com

Static vs. Dynamic pages (cont.)



- Orientations du Web :
  - Services en ligne, E-business, les blog, les forums, etc.
  - Offrir un contenu personnalisé / dynamique qui s'adapte aux besoins de chaque visiteur en fonction de ses actions, + d'interactivité.
- Un page web dynamique est construite à la demande (à la volée) par le serveur (côté serveur), en fonction de critères spécifiques.
- La présentation et le contenu affichés peuvent ainsi être personnalisés de manière interactive, en fonction des produits, des internautes, des langues, etc.

Static vs. Dynamic pages (cont.)



- On reconnaît facilement un page dynamique grâce à l'URL qui s'affiche dans le navigateur web de l'utilisateur:
  - Page statique

    http://www.monsite.fr/accueil.html: affiche la page ***accueil.html***, stockée telle quelle sur le serveur,
  - Page dynamique

    http://www.monsite.fr/accueil.php?langue=en: affiche la page ***accueil.php*** en demandant au serveur d'afficher le contenu de cette page en français.

neila.benlakhal@gmail.com

Static vs. Dynamic pages (cont.)

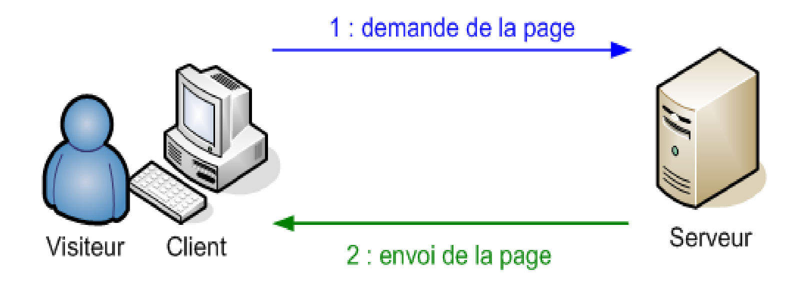


- **Alors que les pages statiques font appel seulement à HTML, les pages dynamiques sont mises en œuvre grâce à des langages de programmation WEB.**
  - But : disposer d'instructions conditionnelles, de boucles et de fonctions de traitement complexes. Le langage de programmation variera en fonction de la technologie retenue (PHP, ASP.NET, JSP, JSF, Perl, etc.).
- **Le langage de programmation Web ne remplace pas le HTML, mais il en produit.**
- **Comment ?**

Statique Vs Dynamique

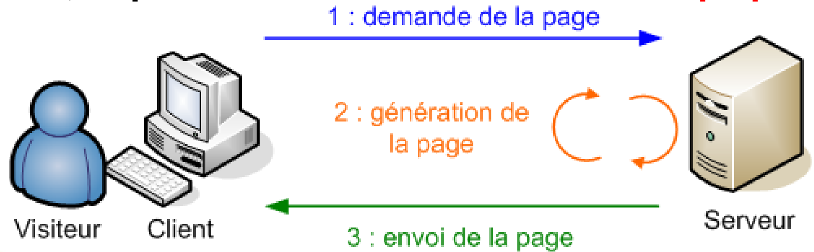


Page Statique
http://www.monsite.fr/accueil.html



- Demander au DNS l'adresse IP correspondant à www.monsite.fr
- Connecter à cette adresse IP sur le port 80
- Demander au serveur la page accueil.html

Page Dynamique
(http://www.monsite.fr/accueil.php



- Le client: "S'il te plaît, envoie-moi la page **accueil.php**".
- Le serveur n'envoie PAS de suite la page au client. Il la **génère** à partir du code php. En effet, le client n'est pas capable d'exécuter une page PHP (seul le serveur sait le faire).
- Une fois la page est générée, le serveur l'envoie au client.

Programmation de pages Web dynamiques : 2 types



- Plusieurs langages sont utilisés pour amener des pages personnalisées aux utilisateurs.

- Type 1 : Programmation coté client (client-side scripting)
  - Changer l'interface de la même page Web, en réponse à un événement bien particulier (entre dans le contexte de la programmation événementielle).
  - Le script (programme) est téléchargé avec la page Web et s'exécute sur la machine du **client**.
  - Exp.: **Javascript**, **VBScript** (seulement sur IE).

Programmation Web : 2 types



- Type 2 : Programmation coté serveur (server-side scripting):
  - Changer le contenu d'une page source et l'ajuster selon la réponse du serveur reçu par le navigateur. La réponse du serveur dépend de:
    - Les données postes dans la requête : HTML forms, paramètres dans l'adresse URL, etc.
  - Dans le monde des logiciels libres, il s'agit souvent de PHP pour le langage et MySQL pour la BD, le langage Perl étant de moins en moins utilisé, JSP(ORACLE previously Sun), etc.
  - Les solutions propriétaires, par exemple celle de Microsoft, avec ASP et Access/MS-SQL Server, ASP.NET.

L'évolution du Web



- Depuis son apparition le Web est passé, du Web 1.0, au Web 2.0 au Web 3.0 (en cours).

- Web 1.0: 1990~2000
  - La version « read-only » du Web
  - Permet essentiellement la recherche et la lecture d'information
  - Très peu de communication entre les utilisateurs ou de contribution au contenu
  - But des utilisateur: être présent sur le Net.

- Web 2.0 : le terme a été proposé en 2003 par Mr. Dale Dougherty, vice-president à O'Reilly Media.



L'évolution du Web

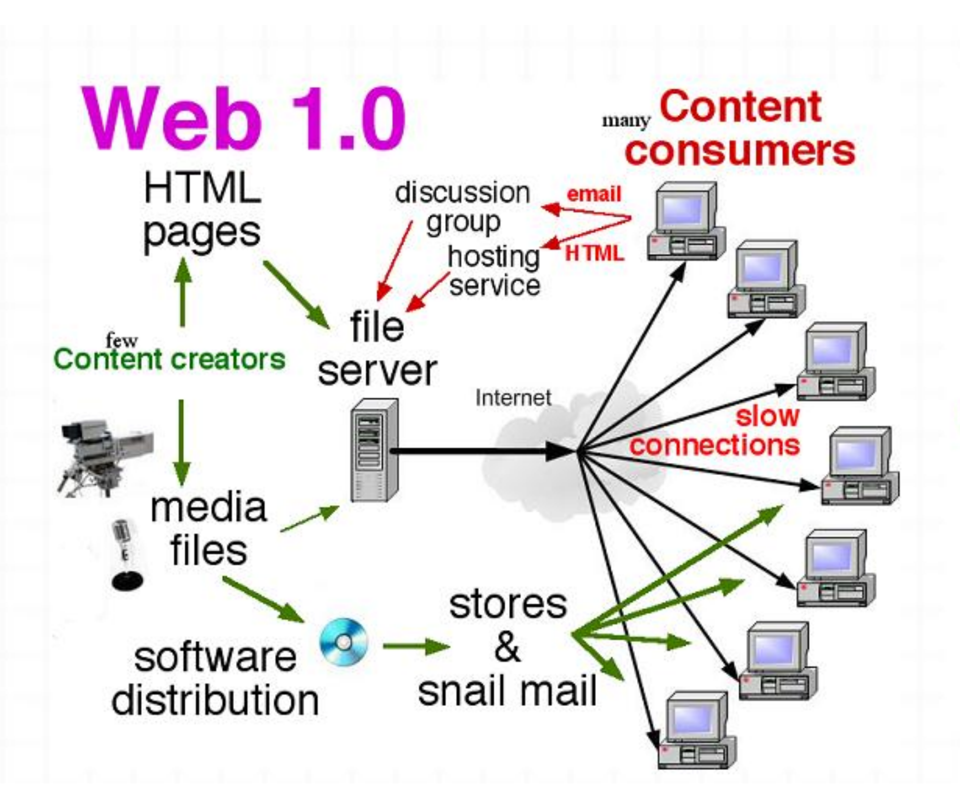


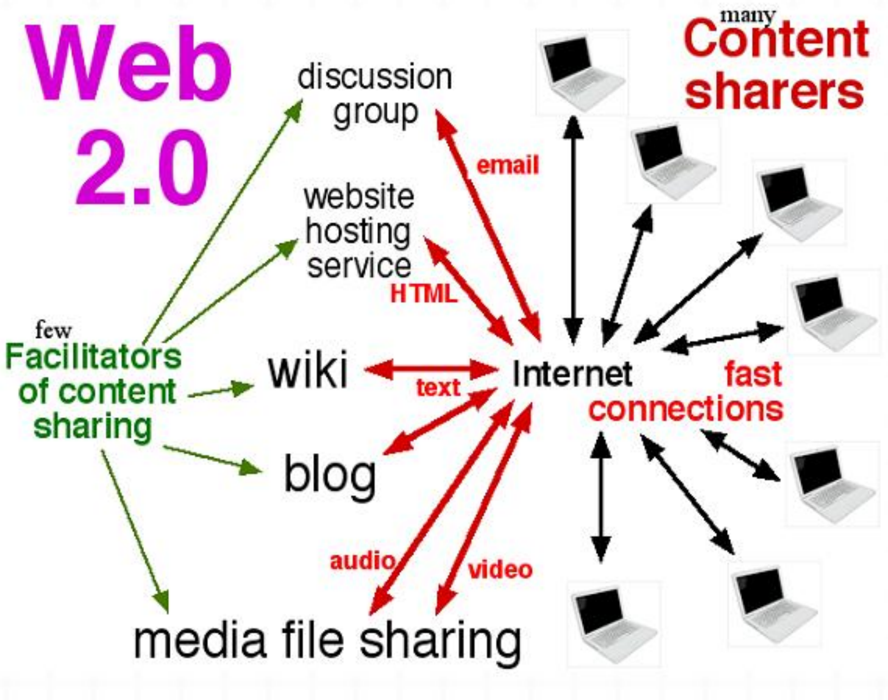
- Web 2.0: 2000~??
  - La version « read-write du Web »
  - Une interaction et une contribution accrue des utilisateurs au contenu du Web
  - Le shift avec l'apparition de Youtube, Myspace
  - Le grand boom avec l'apparition des social networks



Web 1.0 vs Web 2.0







neila.benlakhal@gmail.com

Des exemples de service du Web 2.0



- Blogs, social networks, podcasts, wikis, media sharing sites, …
- Et le Web 3.0 c'est pour quand ?
  - Une décennie pour avoir le Web 2.0
- Le Web 3.0 sera comment selon vous?

L'évolution du Web



- Certains disent que le Web 3.0 pour 2015
  - **Web intelligent**
    - Convergence avec le domaine de l'IA
  - **Le Web sémantique**
    - Les ordinateurs comprendraient l'information qui circulent sur le Web tout autant que les humains

- Certains prédisent qu'il y aura **pas** de Web 3.0!!!
- **Et vous qu'est-ce que vous en pensez ??**

Les grands axes du cours



- HTML4.01 (Rappel ??), HTML5
- CSS1, CSS2, CSS3 (avant gout)
- Client-side scripting: Javascript (librairies), DOM, les APIs de HTML5
- Server-side scripting PhP4/Php5, les frameworks.
- Gestion d'état
- AJAX, JSON, XML
- SimpleXML
- Nouvelles orientations: programmable web, les Mashup, Google app scripting, les promesses de HTML5, le WEB3,0.

Références



- W3C : http://www.w3schools.com/
- **Internet &World Wide Web: How to Program (4th Edition),** Harvey M. Deitel and Paul J. Deitel,, 1400pp., paper (0-13-175242-1) 2008
- **HTML5: Up and Running, Dive into the Future of Web Development**, Mark Pilgrim, O'Reilly Media, 222pp, 2010.
- **AJAX, Rich Internet Applications, and Web Development for Programmers,** Paul J. Deitel, Harvey M. Deitel, Prentice Hall, 1040pp, 2008.
- **PHP6 and MySQL Bible**, Steve Suehring, Tim Converse, Joyce Park, Wiley, Jan 20, 2009

(voir lien du cours pour télécharger une version électronique des livres)

Outils de travail



- éditeurs:
  - Aptana studio 2.0,3,0 (http://aptana.com/)
  - Notepad++ (http://notepad-plus-plus.org/)
  - Amaya (http://www.w3.org/Amaya/)
  - Sublime Text 2 http://www.sublimetext.com/2
  - Microsoft Visual Studio
  - Eclipse Web Tools Platform (WTP) Project Development
  - Adobe Dreamweaver CS6 (payant)
  - Les lives IDE (Google Store, Ex platform, Cloud9, etc..)
  - Etc.
- Validator
  - HTML validator : http://validator.w3.org/
  - CSS validator : http://csslint.net/
  - JS Checker: http://jslint.com/



HTML4.01

Le HTML : Définition de l'acronyme



- ***HTML : <u>H</u>yper<u>T</u>ext <u>M</u>arkup <u>L</u>anguage:*** est un langage de balisage de **description** du **contenu** de pages Web.



- ***Hypertext*** réfère au fait que les pages contiennent bien plus que du texte seulement:
  - Peut contenir des **images**, des **objets multimédia**, des **liens** référant à d'autres partie de la même page/d'autres pages.

- ***Markup*** réfère ici que, autre le texte à afficher, une page comporte des ***balises*** (***tags***) pour définir la **structure** et le **contenu** d'une page.

neila.benlakhal@gmail.com

Les versions du HTML en quelques dates



- **HTML 1.0 (Berners-Lee, 1989):** très basique, intégration limitée d'objet multimédia.
- **HTML 2.0 (IETF, 1994):** ont essayé de standardiser cette version, mais plus tard vers 1994-96, Netscape & IE ont ajoute +sieurs dispositifs divergents.
- **HTML 3.2 (W3C, 1996):** ont essayé d'unifier le tout en un unique standard, technologies comme Java applets & streaming video n'ont pas été prise en considération.
- **HTML 4.0 (W3C, 1997):** L'utilisation de frames, des tableaux plus complexes, des améliorations sur les formulaires etc…Mais surtout, cette version permet pour la première fois l'utilisation de feuilles de style CSS.
- **XHTML1.0 (W3C, 2000): HTML 4.01** modifié pour être conforme au standard XML.
- **XHTML 2.0 (W3C, 2006):** Sortie d'une version de travail de la recommandation XHTML 2.0.
- **HTML5 (2009):** Sortie d'une version de travail de la recommandation HTML 5. standard attendu pour 2014.

Un premier exemple : page01.html



- Un document écrit en langage HTML est un fichier texte avec l'extension **.htm**, **.html**, **.shtm**, **.shtml**, **.xhtml**.

C:\wamp\www\demo\Page01.html - Notepad++

Fichier Edition Recherche Affichage Encodage Langage Paramétrage Macro Exécution TextFX Compléments Documents ?

Page01.html

```html
<html>
<!-- je suis un commentaire -->
<head>
<title>Bienvenue!!</title>
</head>
<body>
<a href=" http://www.w3schools.com/ ">
 ceci est un lien vers le site W3C</a>
<h1>ceci est un titre</h1>
<p>ceci est un paragraphe.</p>
<b>ce texte est en gras</b>
<br/>
<i>ce texte est en italique</i>
</body>
</html>
```



HTML est un langage interprété



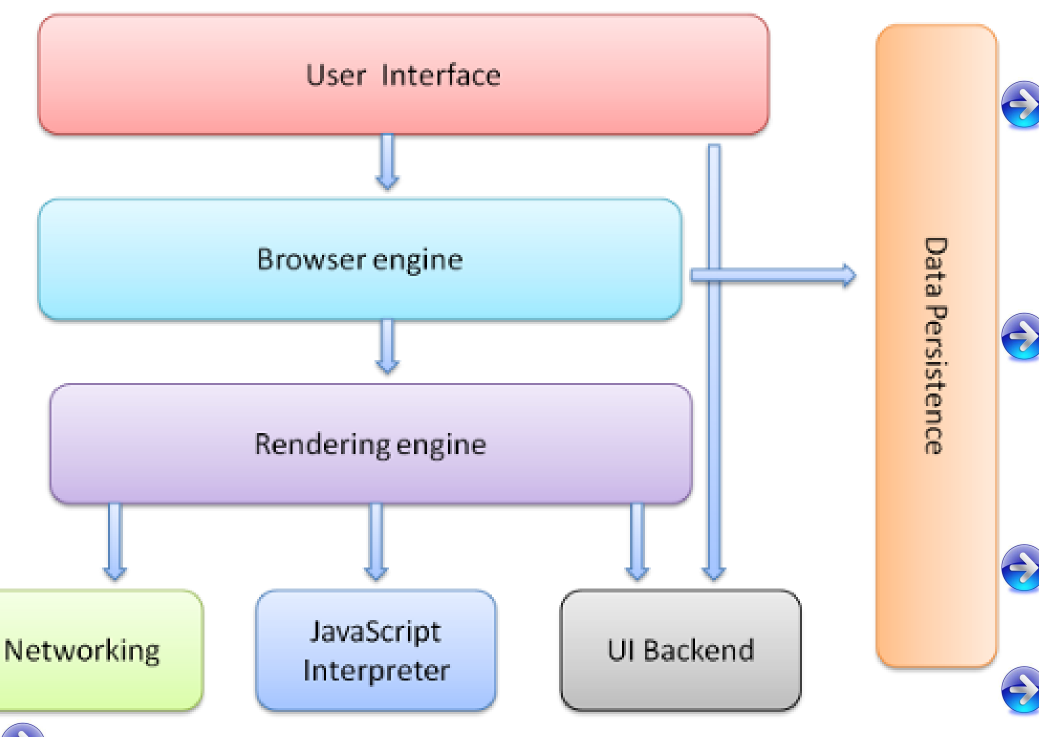
- Les balises sont **invisibles** pour l'internaute :
  - Le navigateur va lire le document HTML et l'afficher sous forme d'une page Web.
  - Le navigateur **n'affiche pas les balises HTML** mais il utilise les balises pour **interpréter** le contenu de la page.
  - HTML **n'est pas** un langage de programmation mais de balisage qui est **interprété** ligne par ligne par le navigateur pour produire le contenu structuré selon les balises utilisées.

neila.benlakhal@gmail.com

Structure de haut niveau d'un navigateur







- **Javascript Interpreter** : utilisé pour parser et éxecuter le code JS
- **Data Storage**: utilisé par le navigateur pour stocker toute sorte d'information persistante comme les cookies etc.
- **User interface (Interface utilisateur)**: bar d'adresse, buttons suivant|précédent, le menu des favoris, etc.
- **Browser engine (Moteur du navigateur) :** l'interface d'envoi des requete au moteur de traitement (Rendering Engine)
- **Moteur de traitement (Rendering Engine)** : responsible de parser le contenu demandé de la page (HTML/CSS) et de l'afficher
- **Networking** : utilisé pour les traitements réseaux (requetes HTTP par exemple)
- **UI Backend:** utilisé pour dessiner à l'écran les fenetres, les boutons de formulaire etc. (utilise le OS)

(fonctionnement détaillé : voir http://taligarsiel.com/Projects/howbrowserswork1.htm)

De quoi est-il constitué page01.html?



- Son contenu est **structuré** à l'aide de repères appelés des **éléments**

```
<title>Bienvenue!!</title>
```

- Chaque élément est constitué de **balises** et d'**attributs** qui permettent d'apporter des informations sur son contenu
- **Balise** : mot-clé du langage HTML entouré par "<"et">".

```
<head>, <body>, …
```

- **Attribut** : un moyen de donner des précisions sur une balise.

```
href, title, src, …
```

C:\wamp\www\demo\Page01.html - Notepad++

Fichier Edition Recherche Affichage Encodage Langage Paramétrage Macro Exécution TextFX Compléments Documents ?

Page01.html

```html
<html>
<!-- je suis un commentaire -->
<head>
<title>Bienvenue!!</title>
</head>
<body>
<a href=" http://www.w3schools.com/ ">
 ceci est un lien vers le site W3C</a>
<h1>ceci est un titre</h1>
<p>ceci est un paragraphe.</p>
<b>ce texte est en gras</b>
<br/>
<i>ce texte est en italique</i>
</body>
</html>
```

leng Ln : 17 Col : 1 Sel : 0 UNIX ANSI

Une balise ça sert à quoi?



- En parcourant un document HTML, les balises servent comme **marqueur** au navigateur pour indiquer :
  - La fonction du texte (lien hypertexte, titre,…) :

```
<a href=" http://www.w3schools.com/ ">ceci est un lien
  vers le site W3C</a>
<h1>ceci est un titre</h1>
```

  - La mise en forme du texte (gras, italique, …):

```
<b>ce texte est en gras</b>
<i>ce texte est en italique</i>
```


neila.benlakhal@gmail.com

Les parties d'un document HTML 4.01 1/2



- La balise **<html>:** C'est la balise principale qui englobera toute votre page HTML . On ne la ferme qu'en dernier avec **</html>** .

- Les deux parties fondamentales d'un document HTML sont l'en-tête (<head>) puis ensuite dans l'ordre le corps (<body>)
  - **l'en-tête** (<head>) : sert à enregistrer des informations complémentaires (mots-clefs, feuilles de styles applicables, des scripts à utiliser, des chargements annexes, etc.)
  - **Le corps du texte** (<body>) :Il contient tout ce qui **apparaîtra** dans la fenêtre du navigateur.

Les parties d'un document HTML4.01 2/2



Le corps : <body>

- Paragraphes
- Titres
- Images
- Liens hypertextes
- Balises de mise en forme
- Listes
- Tableaux

- Formulaire
- Objets multimédia
- Cadres (frames)

L'en-tête : <head>

- Title
- Link
- Script
- style
- Meta

neila.benlakhal@gmail.com

Syntaxe d'un élément HTML



- Un élément commence par une **balise d'ouverture** et se termine par une **balise de fermeture**

  Syntaxe :
  ```
  <balise> contenu </balise>
  ```
  Exemple :
  ```
  <p>ceci est un paragraphe.</p>
  ```
- Certains éléments HTML ont **un contenu vide**
- Les éléments vides sont **fermés dans la balise de début**

  Syntaxe :
  ```
  <balise/>
  ```
  Exemple :
  ```
  <br/>
  ```

Attributs HTML



- Les éléments HTML peuvent avoir **des attributs :** fournissent **des informations supplémentaires** sur un élément
- Les attributs sont toujours spécifiés dans la balise de début et sont définis en paire "nom / valeur" comme name = ”value” (double quote) ou encore name=’value’ (simple quote)

Syntaxe :

```
<balise attribut1="val"> contenu </balise>
<balise attribut='valeur' />
```

Exemple : l’attribut href

```
<a href="http://site.fr/page.html"> ceci est un
  lien</a>
```


neila.benlakhal@gmail.com

Vue Arborescente & Représentation imbriquée



- Les éléments d'un documents HTML sont imbriquées les uns dans les autres pour définir la structure du document :

```
<racine>
<premier_niveau>
<second_niveau> <niveau_3 /></second_niveau>
</premier_niveau>
<autre_balise_au_niveau_1 />
</racine>
```

Par exemple,

```
<html>
<body>
<p>ceci est un paragraphe.</p>
</body>
</html>
```


neila.benlakhal@gmail.com

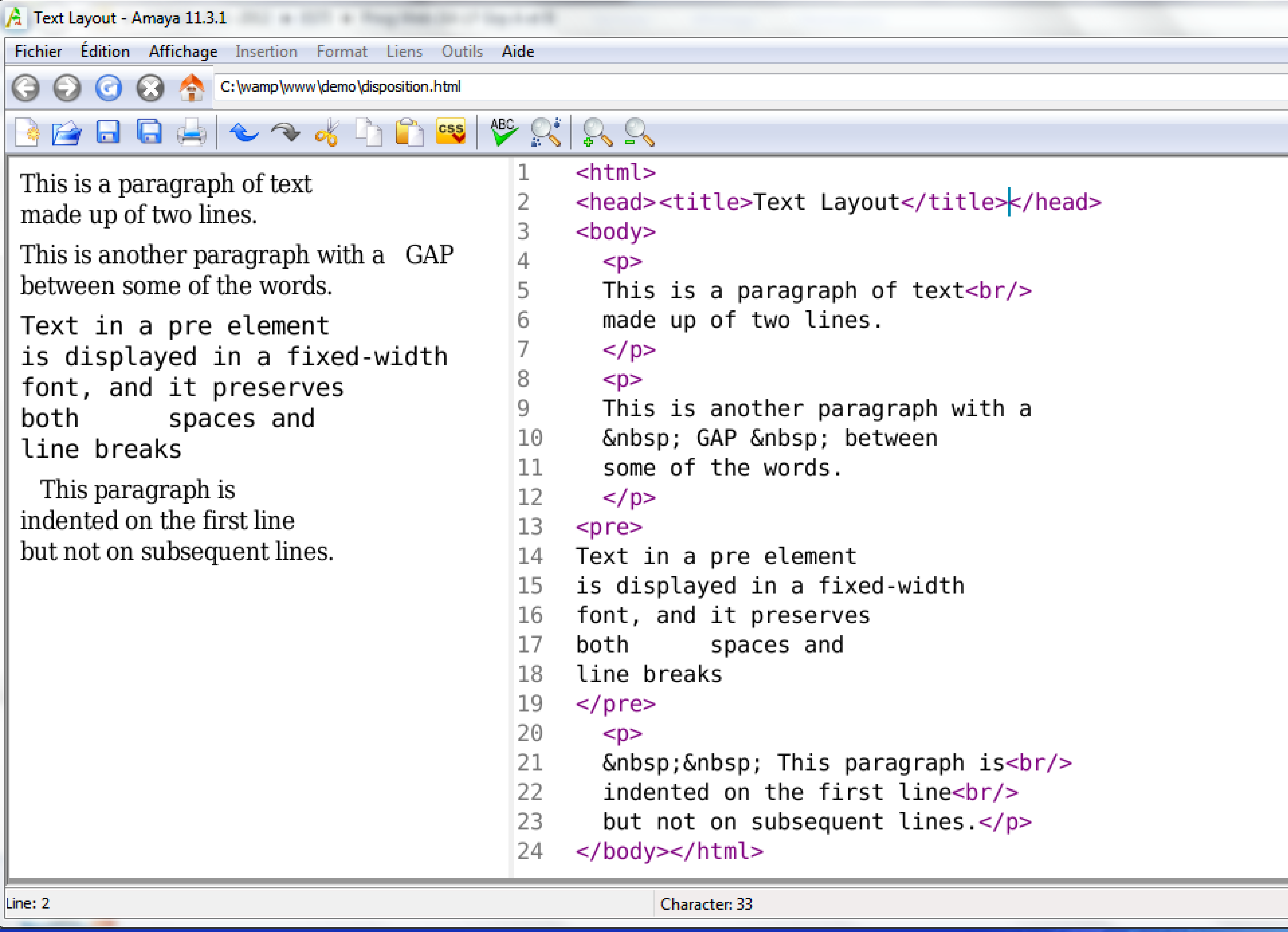
```
<html>
<head><title>Text Layout</title></head>
<body>
  <p>
  This is a paragraph of text<br/>
  made up of two lines.
  </p>

<p>
  This is another paragraph with a
    GAP   between
  some of the words.
</p>

<pre>
Text in a pre element
is displayed in a fixed-width
font, and it preserves
both       spaces and
line breaks
</pre>
  <p>
     This paragraph is<br/>
  indented on the first line<br/>
  but not on subsequent lines.</p>
</body></html>
```

Disposition du texte

- Généralement, la disposition du texte doit être laisser à la charge du navigateur:
  - +sieurs espaces sont interprétés comme un seul espace. Les retours à la ligne sont fais automatiquement par le navigateur.
- HTML donne la possibilité de forcer la disposition du texte:
  - Un retour a la ligne: la balise **
**
  - Un nouveau paragraphe (laisser une ligne vide et retour a la ligne) en utilisant **<p>…</p>**
  - Ajouter un espace dans un emplacement supposé être sans espace: le symbole ** **
  - Utiliser la balise <pre> pour préserver les espaces et les retours à la ligne.



Text Layout - Amaya 11.3.1
Fichier Édition Affichage Insertion Format Liens Outils Aide
C:\wamp\www\demo\disposition.html
This is a paragraph of text
made up of two lines.
This is another paragraph with a GAP
between some of the words.
Text in a pre element
is displayed in a fixed-width
font, and it preserves
both spaces and
line breaks
This paragraph is
indented on the first line
but not on subsequent lines.
<html>
<head><title>Text Layout</title></head>
<body>
<p>
This is a paragraph of text
made up of two lines.
</p>
<p>
This is another paragraph with a
  GAP   between
some of the words.
</p>
<pre>
Text in a pre element
is displayed in a fixed-width
font, and it preserves
both spaces and
line breaks
</pre>
<p>
   This paragraph is
indented on the first line
but not on subsequent lines.</p>
</body></html>
Line: 2
Character: 33

L'élément paragraphe

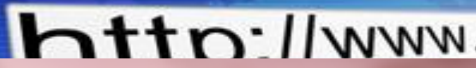


- L'élément **<p>** définit un paragraphe dans le document HTML.
  - Il a une balise de début **<p>** et une balise de fin **</p>**.
  - **<p>**provoque un saut de ligne avant le paragraphe.
  - Un paragraphe ne peut en contenir un autre.

- Attribut principal de **<p>**:

Attribut	Valeur(s)
align	left(par défaut) right center justify

```
<html><head></head>
<body>
  <p>This is a paragraph of
text<br/>made up of two lines.</p>
  <p align="center">This is a
centered paragraph.</p>
</body>
</html>
```



page02.html - Amaya 11.3.1

Fichier Édition Affichage Insertion Format Liens Outils Aide

http://localhost/demo/page02.html

Bienvenue à Ama… | page02.html

This is a paragraph of text
made up of two lines.

This is a centered paragraph.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
      "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
<body>
<p>This is a paragraph of text<br/>  made up of two lines.</p>
<p align="center">This is a centered paragraph.</p>
</body>
</html>
```

Titre et sous-titre




```
<html>
<body>
<h1>Titre super important</h1>

<h2>Titre important</h2>

<h3>Titre un peu moins
important</h3>

<h4>Titre pas trop important</h4>

<h5>Titre pas important</h5>

<h6>Titre vraiment pas important du
tout</h6>

</body>
</html>
```

- Il s'agit ici des titres qui s'afficheront dans la page : ce sont les en-têtes de sections ou paragraphes.
- Il existe 6 niveaux de titres d'importance décroissante :
  - **<h1> </h1>** : signifie "titre très important". En général, on s'en sert pour afficher le titre de la page en haut.
  - **<h2> </h2>** : signifie "titre important".

    …
  - **<h6> </h6>** : titre vraiment pas important du tout.



titre et sous-titre - Amaya 11.3.1

Fichier Édition Affichage Insertion Format Liens Outils Aide

C:\wamp\www\demo\Page03.html

page02.html | titre et sous-t...

Titre super important

Titre important

Titre un peu moins important

Titre pas trop important

Titre pas important

Titre vraiment pas important du tout

```
<html>
<head>
<title>titre et sous-titre</title>
</head>
<body>
<h1>Titre super important</h1>
<h2>Titre important</h2>
<h3>Titre un peu moins important</h3>
<h4>Titre pas trop important</h4>
<h5>Titre pas important</h5>
<h6>Titre vraiment pas important du tout</h6>
</body>
</html>
```

Line: 3 Character: 12

Lien hypertexte 1/4



- Un hyperlien (ou lien) est un mot, un groupe de mots, ou une image que vous pouvez cliquer dessus pour passer à un autre page Web ou pour passer à une nouvelle section de la même page.

- Un lien est défini en utilisant la balise <a>qui peut être utilisé pour créer :
  - Lien externe: vers un autre document, en utilisant l'attribut href
  - Lien interne/signet/ancre dans un même document, en utilisant l'attribut name

Lien hypertexte 2/4



Syntaxe :

```
<a href="url" target="_blank">lien externe</a>
```

- L'attribut *href* spécifie la destination d'un lien.
- L'attribut *target* spécifie où ouvrir le document destination.

Exemple :

```
<a href=" http://www.w3schools.com/ "> ceci est un lien vers le site du W3C</a>
```

Bienvenue!! - Amaya 11.3.1

Fichier Édition Affichage Insertion Format Liens Outils Aide

http://localhost/demo/Page01.html

ceci est un lien vers le site W3C

ceci est un titre

ceci est un paragraphe.

ce texte est en gras
ce texte est en italique

```
<html>
<!--~je suis un commentaire~-->
<head>
<title>Bienvenue!!</title>
</head>
<body>
<a href=" http://www.w3schools.com/ ">
 ceci est un lien vers le site W3C</a>
<h1>ceci est un titre</h1>
<p>ceci est un paragraphe.</p>
```

Lien hypertexte 3/4



- Un **lien interne** pointe vers une ancre, c'est à dire un endroit à l'intérieur d'un document défini par un nom. Il permet de naviguer dans les longs documents.

Syntaxe :

```
<a name="variable" >ancre ici</a>
```

  - L'attribut *name* spécifie le nom d'un ancre.
  - Le lien proprement dit se comme suit :

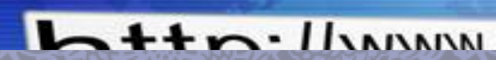
Exemple :

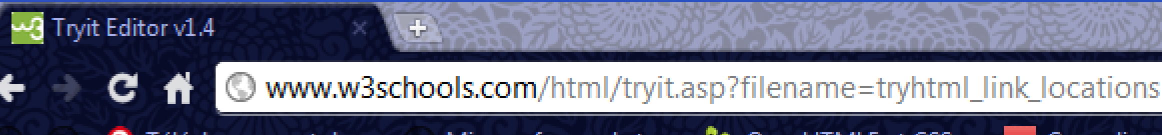
```
<a href="#variable" >lien vers ancre</a>
```

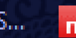
- On peut créer un lien vers la "Partie 1" d'une autre page

```
<a href="#partie1">Première Partie</a>
...<a name="partie1"></a><h1>Partie 1</h1>
```

```
<a href="http://www.site.rnu.tn/liens.html#partie1">
Première Partie</a>
```









Edit and Click Me >>

Your Result:

```
<html>
<body>
<p>
<a href="#C4">See also Chapter 4.</a>
</p>
<h2>Chapter 1</h2>
<p>This chapter explains ba bla bla</p>
<h2>Chapter 2</h2>
<p>This chapter explains ba bla bla</p>
<h2>Chapter 3</h2>
<p>This chapter explains ba bla bla</p>

<h2><a name="C4">Chapter 4</a></h2>
<p>This chapter explains ba bla bla</p>
```

See also Chapter 4

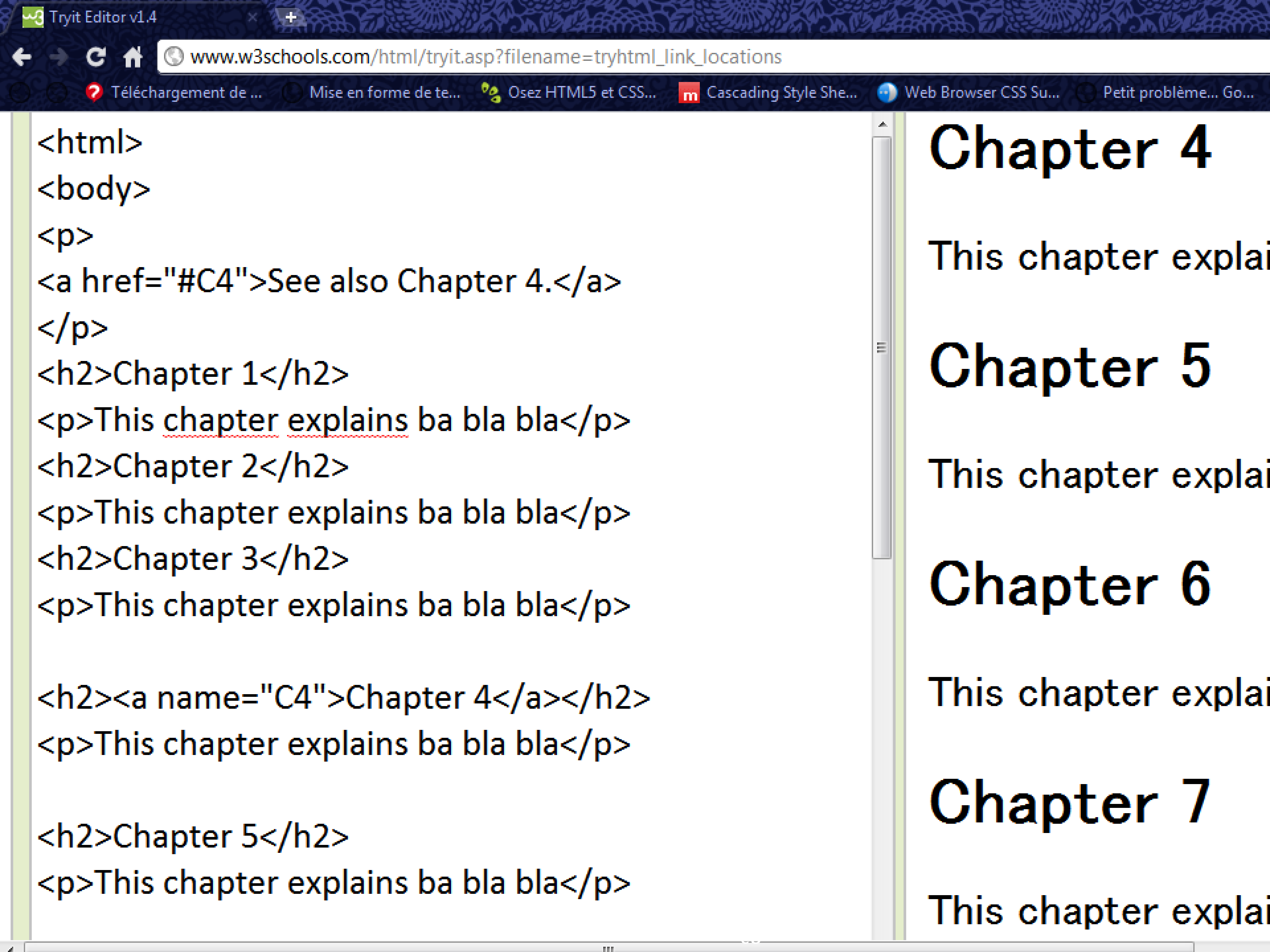
Chapter 1

This chapter explai

Chapter 2

This chapter explai

Chapter 3


Tryit Editor v1.4
www.w3schools.com/html/tryit.asp?filename=tryhtml_link_locations
Téléchargement de ...
Mise en forme de te...
Osez HTML5 et CSS...
Cascading Style She...
Web Browser CSS Su...
Petit problème... Go...
<html>
<body>
<p>
See also Chapter 4.
</p>
<h2>Chapter 1</h2>
<p>This chapter explains ba bla bla</p>
<h2>Chapter 2</h2>
<p>This chapter explains ba bla bla</p>
<h2>Chapter 3</h2>
<p>This chapter explains ba bla bla</p>
<h2>Chapter 4</h2>
<p>This chapter explains ba bla bla</p>
<h2>Chapter 5</h2>
<p>This chapter explains ba bla bla</p>
Chapter 4
This chapter explai
Chapter 5
This chapter explai
Chapter 6
This chapter explai
Chapter 7
This chapter explai

Lien hypertexte 4/4



- Exemples de liens vers des ressources utilisant d'autres protocoles

Lien e-mail sans sujet :

```
<a href="mailto:neila.bl@gmail.fr">contact</a>
```

Lien externe vers un serveur FTP :

```
<a href="ftp://ftp.server.fr">upload link</a>
```

Lien vers d'autres objets : Le système d'exploitation recherchera de lui-même, selon l'extension du fichier, quelle application permet de l'exploiter. Vous pouvez proposer un fichier (.EXE ou .ZIP) en téléchargement...

```
<a href="site.zip">Téléchargement du site</a>
```

Images 1/3



- La balise <img> est le moyen le plus simple d'insérer une image dans une page Web

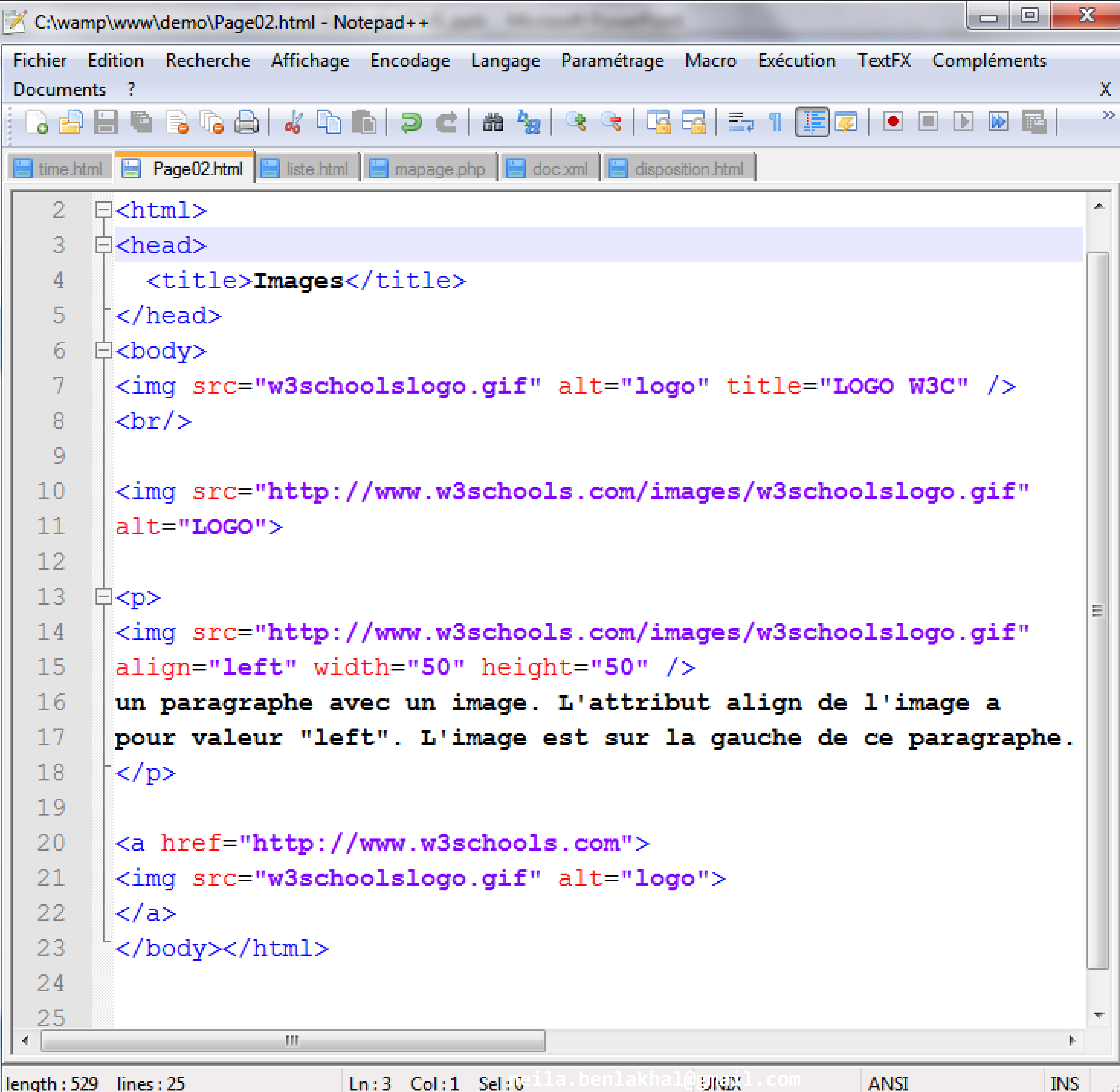
Syntaxe :

```
<img src="url" alt="some_text" title="text"/>
```

- La balise <img> est vide, c.à.d. elle contient uniquement des attributs et pas de balise de fermeture.
- *Src* : l'URL de l'image que vous souhaitez afficher.
- *Alt* : est un texte alternatif à afficher si l'image ne se charge pas.
- *Title* affiche un texte dans une info-bulle.

Exemple :

```
<img src="w3schoolslogo.gif" alt="logo" title="Logo W3C" />
<img src="http://www.w3schools.com/images/w3schoolslogo.gif"
alt="logo" title="logo W3C" >
```


C:\wamp\www\demo\Page02.html - Notepad++
Fichier Edition Recherche Affichage Encodage Langage Paramétrage Macro Exécution TextFX Compléments
Documents ?
time.html Page02.html liste.html mapage.php doc.xml disposition.html
<html>
<head>
<title>Images</title>
</head>
<body>

<img src="http://www.w3schools.com/images/w3schoolslogo.gif"
alt="LOGO">
<p>
<img src="http://www.w3schools.com/images/w3schoolslogo.gif"
align="left" width="50" height="50" />
un paragraphe avec un image. L'attribut align de l'image a
pour valeur "left". L'image est sur la gauche de ce paragraphe.
</p>

<img src="w3schoolslogo.gif" alt="logo">

</body></html>
length : 529 lines : 25 Ln : 3 Col : 1 Sel : 0 UNIX ANSI INS
leila.benlakhal@gmail.com

Images 2/3



Images

localhost/demo/Page02.html

Ecouter Shems FM e... Téléchargement de ... Mise en forme de te... その他のブックマーク

w3schools.com

w3schools.com

un paragraphe avec un image. L'attribut align de l'image a pour valeur "left". L'image est sur la gauche de ce paragraphe.

w3schools.com

Images 3/3



- Le navigateur affiche l'image où la balise *<img>* se trouve.
- Si vous placez la balise *<img>* entre 2 paragraphes, le navigateur affiche l'image entre le 1er et le 2ème paragraphe.
- Utilisez l'attribut *align* pour laisser une image flottante à gauche ou à droite d'un paragraphe :

```
<p>
<img src="http://www.w3schools.com/images/w3schoolslogo.gif"
align="left" width="50" height="50" />un paragraphe avec un image.
L'attribut align de l'image a pour valeur "left". L'image est sur
la gauche de ce paragraphe.</p>
```

- Une image comme lien hypertexte :

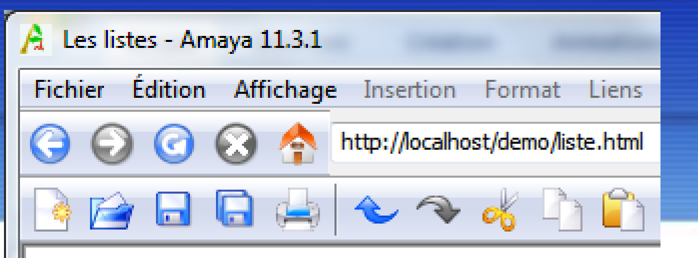
```
<a href="http://www.w3schools.com">
<img src="w3schoolslogo.gif" alt="logo">
</a>
```

Les listes



La notion de liste permet de structurer un ensemble de données :

- Type 1 : Listes non ordonnées (**U**nordred **L**ist) :
  - Une liste non ordonnée commence par la balise **‹ul›**.
  - Chaque élément de la liste commence avec la balise **‹li›.**
  - Il y aura autant de **‹li›** que d'élément dans la liste.
- Type 2 : Les listes numérotées (**O**rdered **L**ist)
  - Une liste non ordonnée commence par la balise **‹ol›**.
  - Chaque élément de la liste commence avec la balise **‹li›.**
- Type 3 : Les listes de définition (**D**efinition **L**ist)
  - La balise **‹dl›** définit une liste de définitions.
  - La balise **‹dt›** (définit l'élément de la liste) et **‹dd›** (décrit l'élément de la liste):





ordered list

1. Je me leve
2. Je mange et je bois
3. Je retourne me coucher

unordered list

- Fraises
- Framboises
- Cerises

list of definition

Hacker
: a clever programmer

Nerd
: technically bright but socially inept person

Line: 11

Les listes : un exemple 1/2




```
<html> <head> <title>Les listes</title> </head><body>
<h1> ordered list</h1>
<ol>
<li>Je me leve</li>
<li>Je mange et je bois</li>
<li>Je retourne me coucher</li>
</ol>
<h1> unordered list</h1>
<ul>
<li>Fraises</li>
<li>Framboises</li>
<li>Cerises</li>
</ul>
<h1>list of definition</h1>
<dl>
<dt> Hacker </dt>
<dd> a clever programmer </dd>
<dt> Nerd </dt>
<dd> technically bright but socially inept person </dd>
</dl>
</body>
</html>
```


neila.benlakhal@gmail.com

Les listes - Amaya 11.3.1

Fichier Édition Affichage Insertion Format Liens Outils Aide

http://localhost/demo/liste.html

ordered list

1. Je me leve
2. Je mange et je bois
3. Je retourne me coucher

unordered list

- Fraises
- Framboises
- Cerises

list of definition

Hacker
: a clever programmer

Nerd
: technically bright but socially inept person

```html
<html>
<head><title>Les listes</title></head>
<body>
<h1>ordered list</h1>
<ol>
<li>Je me leve</li>
<li>Je mange et je bois</li>
<li>Je retourne me coucher</li>
</ol>
<h1>unordered list</h1>
<ul>
<li>Fraises</li>
<li>Framboises</li>
<li>Cerises</li>
</ul>
<h1>list of definition</h1>
<dl>
<dt>Hacker</dt>
<dd>a clever programmer</dd>
<dt>Nerd</dt>
<dd>technically bright but socially inept person</dd>
</dl>
</body>
</html>
```

Line: 11 Character: 1 Texte 

Les tableaux



- En HTML, un tableau est un conteneur définie par la balise **‹table›** et constitué de lignes **‹tr›** (pour ***Table Row***) et de cellules **‹td›** (pour ***Table Data***)

- Un tableau est divisé en lignes (avec les **‹tr›**), et chaque rangée est divisée en cellules de données (avec les **‹td›**).
  - ‹td› peut contenir du texte, liens, images, listes, formulaires, d'autres tables, etc.

- Par default, le contenu des colonne est **justifié a gauche**, et **sans bordure** pour cela il faut définir un alignement et les bordures **explicitement**.

Les tableaux: exemple




```
<html>
<head><title>Tables</title>
</head>
<body>
<h2>A Simple Table</h2>
  <table>
   <tr>
     <td> Left Column </td>
     <td> Right Column </td>
   </tr>
   <tr>
     <td> Some data </td>
     <td> Some data </td>
   </tr>
 </table>
</body>
</html>
```

A Simple Table

Left Column	Right Column
Some data	Some data

neila.benlakhal@gmail.com

Les tableaux



Bordure :

- Si vous ne spécifiez pas un attribut de bordure (***border***), le tableau sera affiché sans bordure.

Header 1	Header 2
row 1, cell 1	row 1, cell 2
row 2, cell 1	row 2, cell 2

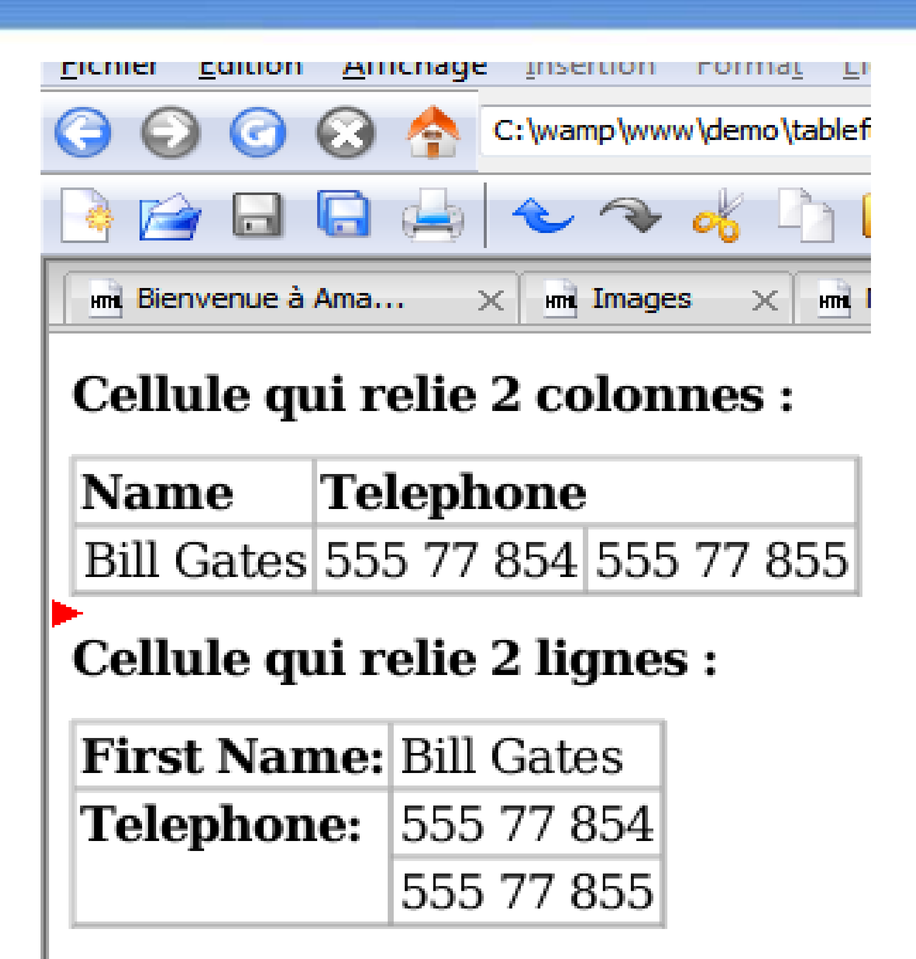
```

<table border="1">
<tr>
<th>Header 1</th>
<th>Header 2</th>
</tr>
<tr>
<td>row 1, cell 1</td>
<td>row 1, cell 2</td>
</tr>
<tr>
<td>row 2, cell 1</td>
<td>row 2, cell 2</td>
</tr>
</table>
```

En-tête:

- Définis avec la balise ***‹th›***.
- Le texte dans un élément sera en gras et centré.





neila.benlakhal@gmail.com

Tableaux : les attributs rowspan et colspan



Cellule qui relie 2 colonnes :

Name	Telephone	
Bill Gates	555 77 854	555 77 855

Cellule qui relie 2 lignes :

First Name:	Bill Gates
Telephone:	555 77 854
	555 77 855

- **rowspan** : Relier des cellules dans une rangée sur plusieurs colonnes
- **colspan** : Relier des cellules dans une colonne sur plusieurs rangées

```html
<html><body>
<h4>Cellule qui relie 2 colonnes :</h4>
<table border="1">
<tr>
  <th>Name</th>
  <th colspan="2">Telephone</th>
</tr>
<tr>
  <td>Bill Gates</td>
  <td>555 77 854</td>
  <td>555 77 855</td>
</tr>
</table>
<h4>Cellule qui relie 2 lignes : </h4>
<table border="1">
<tr>
  <th>First Name:</th>
  <td>Bill Gates</td>
</tr>
<tr>
  <th rowspan="2">Telephone:</th>
  <td>555 77 854</td>
</tr>
<tr>
  <td>555 77 855</td>
</tr>
</table>
```

Mis en en forme



- HTML4.01 défini plusieurs balises pour changer la mise en forme du texte :

Balise	Description
<b>	texte en gras
<big>	taille supérieure à la taille courante
<em>	accentuation
<i>	texte en italique
<small>	taille inférieur à la taille courante
<strong>	Plus d'accentuation
<sub>	texte en indice
<sup>	texte en exposant
<ins>	Définit le texte inséré
<del>	Définit le texte supprimé

Mis en en forme : exemple



Text Variations

We can use **simple** tags to *change* the appearance of**text** within `Web pages`. Even super^{script} and sub$_{\text{scripts}}$ are *supported*.

```
<html>
<head>
<title>Text Variations and Escape Sequences</title>
</head>

<body>
<h1>Text Variations</h1>

<p>We can use <b>simple</b> tags to <i>change</i>
the appearance of<strong>text</strong> within <tt>Web pages</tt>. Even super<sup>script</sup>
and sub<sub>scripts</sub> are <em>supported</em>.</p>
```

Mise en forme : autres balises (html 4.01)

http://www.

- Autres balise de mise en forme !
  - Computer Tags : <tt>,<pre>, <code>, <samp> …
  - Citation, définition, etc : <abbr>, <blockquote>,…
  - Direction du texte,
  - Etc..

Les entités HTML



- Elles sont utilisées pour écrire les caractères spéciaux du langages, les caractères accentuées etc.

```
caractère(s)   entité


< >            &lt; &gt;
é è            &eacute; &egrave;
™ ©            &trade; &copy;
π δ Δ          &pi; &delta; &Delta;
" &            &quot; &amp;
```

- Pour une liste complète : http://www.w3schools.com/tags/ref_entities.asp

neila.benlakhal@gmail.com

Pourquoi écrire du code HTML Valide et respectant les standards du Web ?



- Dans le développement Web, il est essentiel que votre code HTML soit correct qui respecte minutieusement la syntaxe du Language :
  - Pour garantir la bonne interprétation de votre code sur différent navigateurs
  - Pour augmenter la chance que votre code va être correctement affiché avec les versions futurs des navigateurs
  - Pour garantir que votre code peut être interchangeable et/ou imbriqué avec d'autre Languages
    - Exp. XML, MathML, JS, PhP, etc.

neila.benlakhal@gmail.com

Importance du balisage



- Un document HTML est **'bien formé'** au sens du W3C si
  - Toute balise ouverte est fermée
  - L'ordre de fermeture respecte l'ordre inverse d'ouverture
  - Les balises sont écrites en minuscule. Les versions futures de HTML ne vont plus acceptées les balises en MAJ.
- Comment savoir si une page est bien formée ?

neila.benlakhal@gmail.com

HTML Tidy et W3C Validator



- **HTML Tidy**, ou plus simplement **Tidy**, est un logiciel permettant de corriger et de valider le code HTML d'une page Web selon les critères émis par le W3C.
  - Après analyse de la syntaxe d'un fichier HTML, il corrige et valide celui-ci. Il est capable de corriger plusieurs erreurs, mais lorsque la correction lui semble trop difficile à déterminer, il envoie un message d'avertissement.
  - Il existe d'autres versions de ce logiciel avec une interface graphique plus pratique, comme TidyGUI.
- Pour plus d'information sur HTML Tidy:

http://infohound.net/tidy/

W3C Validator



- Ajouter à votre page lien hypertexte suivant vers la page du w3C validator

```
<p> <a
href="http://validator.w3.org/check/referer">
<img src="http://www.w3.org/Icons/valid-
html401" alt="Validate" /> </a> </p>
```

- Le W3C validator validera pour vous automatiquement la page de la quel le lien a été cliqué

W3C XHTML 1.0 W3C HTML 4.01 W3C HTML 5

neila.benlakhal@gmail.com

W3C online validator



validator.w3.org/#validate_by_uri+with_options

Web 2.0 Icons - Dow... | ..:: calibre library ::.. ... | COMPT1 | Flash Jigsaw | 0 Notifications | MOSAIQUE FM :: LIVE | Télécharger - Sams ... | Ebooks Livres - Ergor | その他のブックマーク

Markup Validation Service
Check the markup (HTML, XHTML, ...) of Web documents

Validate by URI | Validate by File Upload | Validate by Direct Input

Validate by URI

Validate a document online:

Address:

▾ More Options

Character Encoding (detect automatically) ☐ Only if missing

Document Type (detect automatically) ☐ Only if missing

◉ List Messages Sequentially ○ Group Error Messages by Type

☐ Show Source ☑ Clean up Markup with HTML-Tidy

☐ Show Outline ☐ Validate error pages ☐ Verbose Output

Check

This validator checks the markup validity of Web documents in HTML, XHTML, SMIL, MathML, etc. If you wish to validate specific content such as RSS/Atom feeds or CSS



Les formulaires HTML (4.01)

neila.benlakhal@gmail.com

Les Formulaires



- Dans des formulaires, l'utilisateur peut compléter des champs de saisie, faire des choix dans des listes et cliquer sur des boutons etc.
- Il faut associer un traitement selon le besoin :
  - Un traitement sur le client, avec Javascript par exemple.
  - Un traitement sur le serveur, avec PHP par exemple.
- Dans quel but ?
  - Récolter de l'utilisateur des renseignements.
  - Permettre a recherche dans des BDs.
  - Offrir à l'utilisateur la possibilité d'une interaction individuelle par exemple en commandant un produit déterminé dans un assortiment de produits.
  - Etc.

Syntaxe



- Balise placée dans la balise ‹body›:

Syntaxe: **‹form method="" action="" name="" ›…‹/form›**

  - Les champs de type **input**, **select**, ou **textarea** sont placées à l'intérieur de l'élément **form**

- Attributs:
  - **method**: valeurs **GET** ou **POST** qui indique la façon dont les données du formulaire sont transmises au serveur.
  - **action (obligatoire)**: nom de la page qui sera exécuté quand l'utilisateur clique sur un bouton de soumission.
  - **name**: le nom du formulaire.

Exemple :

```
<form action="form_action.php" method="get"
name="f1">
…</form>
```

Les éléments d'un formulaire





Form Test

- Simple inputs :
  - Text :
  - Password :
  - Radio : Choice 1 Choice 2 Choice 3
  - Checkbox : Box 1 Box 2 Box 3
  - Hidden :
- Selection Fields :
  - Simple : Selection Value 1
  - Disabled : Selection Value 1
  - Preselected : Selection Value 2
  - Multiple : Selection Value 1 / Selection Value 2
- Text area :
- File upload : Parcourir...
- Validation button :
  - Submit : Envoyer
  - Reset : Effacer

3 catégories :

- **Input** : Champs de saisie de texte + les boutons
  - type ="text" – zone de texte (type par défaut)
  - type ="password" – zone de texte caché
  - type ="radio" – minimum 2, un seul sélectionnable
  - type ="checkbox" – cases à cocher
  - type = "submit" – soumission de formulaire
  - type = "reset" – bouton de remise à zéro des champs
  - type ="button" – bouton associé à du code Javascript
  - type="hidden" – bouton caché
- **Select**: menus déroulants, listes à choix
  - Size="1"- liste simple, 1 seul élément sélectionnable
  - Size="1"- liste à choix multiple
- **Textarea**: zone de saisie d'un texte "long"

Zone de saisie texte



Syntaxe :

```
<input type="text" size=".." maxlengh=".." name="nom" value="valeur par défaut"/>
```

Les attributs :

- **size**: taille d'affichage de la zone en caractère, 20 par défaut.
- **disabled="disabled"** (non utilisable et non cliquable)
- **readonly="readonly"** (non modifiable)
- **Maxlengh**=le nombre maximal de caractère permis

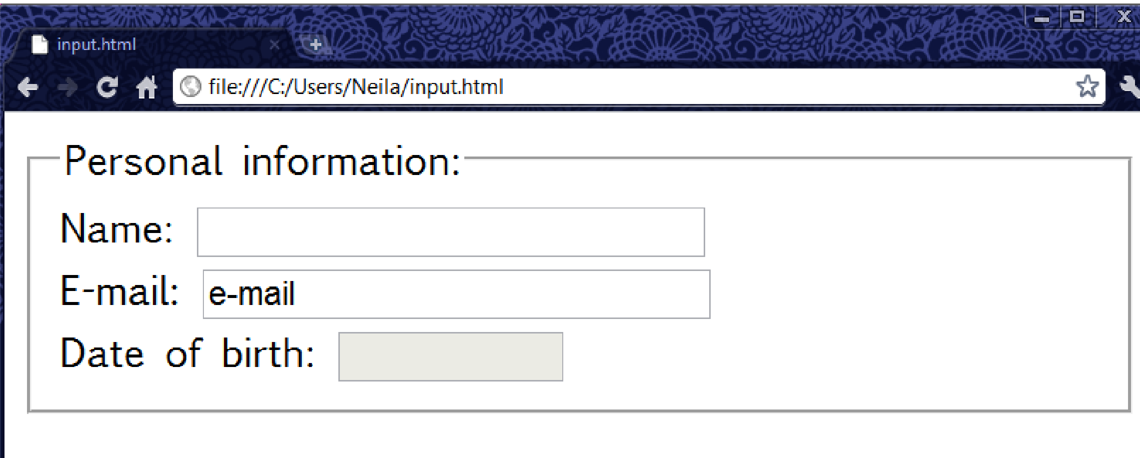
Exemple :

```
<html>
<body>
<form><fieldset><legend>Personal information:</legend>
   <label for="Name">Name:</label><input type="text" size="30" /><br/>
   <label for="e-mail">E-mail:</label><input type="text" size="30" />
</fieldset></form>
</body>
</html>
```

Zone de saisie texte : exemple




```
<html>
<body>
<form><fieldset>
<legend>Personal information:</legend>
   <label for="Name">Name:</label>
   <input type="text" size="30" maxlength="30" name="fname"/><br/>
   <label for="e-mail">E-mail:</label>
   <input type="text" size="30" readonly="readonly" value="e-mail"/><br/>
   <label for="Date of birth">Date of birth:</label>
   <input type="text" size="10" disabled="disabled" />
</fieldset>
</form>
</body>
</html>
```



Zone de saisie de mot de passe



Syntaxe :

```
<input type="password" size="." maxlengh=".." name="nom" value="valeur par défaut"/>
```

Les attributs :

- **size**: taille d'affichage de la zone en caractère, 20 par défaut.
- **disabled="disabled"** (non utilisable et non cliquable)
- **readonly="readonly"** (non modifiable)
- **Maxlength**=le nombre maximal de caractère permis. Généralement size=maxlength.

Exemple :

```
<html>
<body><form><fieldset>
<legend>Personal information:</legend>
   <label for="Password">Password:</label>
   <input type="password" size="8" name="fpass"/>
</fieldset>
</form>
</body>
</html>
```

Zone de saisie de mot de passe: exemple




```
<html>
<body>
<form action=""><fieldset>
<legend>Personal information:</legend>
   <label for="Name">Name:</label>
   <input type="text" size="30" maxlength="30" name="fname"/><br/>
   <label for="Password">Password:</label>
   <input type="password" maxlength="8" value="pass" name="fpass"/><br/>
   </fieldset>
</form>
</body>
</html>
```

input.html

file:///C:/Users/Neila/input.html

Personal information:

Name:

Password: ••••

Définir une liste de choix



- Cases à cocher permettant un choix multiple :

Syntaxe :
```
<input type="checkbox" name="nom" value="val"/>
```

- Attributs :
  - Name, value, disabled="disabled" (non sélectionnable)
  - **checked="checked"** (coché)

Exemple :

```
<html><body><form action="">
<input  type="checkbox" name="langage" value="js" />
<label for>Javascript<label><br />
<input type="checkbox" name="langage" value="php" checked="checked"/>
<label for>php<label></form></body></html>
```

- NB: l'attribut **name** a la même valeur pour relier +sieurs choix!!!

neila.benlakhal@gmail.com

Définir une liste de choix : exemple




```
<html>
<body>
<form action="">
<input type="checkbox" name="langage" value="js" />
<label for>Javascript<label><br />
<input type="checkbox" name="langage" value="php" checked="checked"/>
<label for>php<label><br />
<input type="checkbox" name="langage" value="java"/>
<label for>java<label><br />
</form>
</body>
</html>
```

checkbox.html

file:///C:/Users/Neila/checkbox.html

☐ Javascript

☑ php

☐ java

Définir des cases d'option



- Choix d'une et une seule option parmi n (appelé aussi boutons radios)

Syntaxe :

```
<input type="radio" name="nom" value="val"/>
```

- Attributs :
  - Name, value, disabled="disabled" (non sélectionnable)
  - **checked="checked"** (coché)

Exemple :

```
<html><body><form action="">
<input  type="radio" name="langage" value="js" />
<label for>Javascript<label><br />
<input type="radio" name="langage" value="php" checked="checked"/>
<label for>php<label></form>
</body></html>
```

Définir des cases d'option: exemple




```
<html>
<body>
<form action="">
<input type="radio" name="langage" value="js" />
<label for>Javascript<label><br />
<input type="radio" name="langage" value="php" checked="checked"/>
<label for>php<label><br />
<input type="radio" name="langage" value="java"/>
<label for>java<label><br />
<input type="radio" name="langage" value="perl" disabled="disabled"/>
<label for>Perl<label><br />
</form>
</body>
</html>
```

checkbox.html

file:///C:/Users/Neila/checkbox.html

Javascript

php

java

Perl

Bouton de validation



- Envoi des données et exécution du programme PHP spécifié par l'attribut **action** de **‹form›**

Syntaxe:

```
<input type="submit" name="nom" value="ok"/>
```

- Attributs :
  - Name, type, size, disabled="disabled" (non sélectionnable)
  - **value**: permet de différencier le traitement à effectuer s'il y a plusieurs boutons.
- Exemple :

```
<form>…
<input type="submit" value="Envoyez" size="40"/>
<input type="submit" value="Envoyez (désactivé)"
disabled="disabled"/>
</form>
```

```
<form name="input" action="html_form_action.php">
<input type="radio" name="langage" value="js" />
<label for>Javascript<label><br />
<input type="radio" name="langage" value="php" checked="checked"/>
<label for>php<label><br />
<input type="radio" name="langage" value="java"/>
<label for>java<label><br />
<input type="radio" name="langage" value="perl" disabled="disabled"/>
<label for>Perl<label><br />
<input type="submit" value="Envoyez" size="40"/>
<input type="submit" value="Envoyez (d&#xe9;sactiv&#xe9;)" disabled="disabled"/>
</form>
<p>Si vous cliquez le bouton "Envoyez", le contenu du formulaire va etre envoy&#xe9; &#xe0; une page appel&#xe9;e "html_form_action.php". Remarquez que l'autre bouton est d&#xe9;sactiv&#xe9;</p>
</body>
</html>
```

checkbox.html

file:///C:/Users/Neila/checkbox.html

Javascript
php
java
Perl

Envoyez | Envoyez (désactivé)

Si vous cliquez le bouton "Envoyez", le contenu du formulaire va etre envoyé à une page appelée "html_form_action.php". Remarquez que l'autre bouton est désactivé

Le bouton reset



- Recharge tous les champs du formulaire à leur valeur par défaut **ET** ne provoque **PAS** l'envoi du contenu à la page PHP associée au formulaire.

Syntaxe:

```
<input type="reset" name="nom" value="annuler"/>
```

- Attributs :
  - name, type, size, disabled="disabled" (non sélectionnable)
  - **value**: permet de différencier le traitement à effectuer s'il y a plusieurs boutons.
- Exemple :

```
<form>…
<input type="reset" value="Annuler" size="40"/>
<input type="reset" value="Annuler (désactivé)"
disabled="disabled"/>
</form>
```


neila.benlakhal@gmail.com

```html
<html>
<body>
<form name="input" action="html_form_action.php">
<input type="radio" name="langage" value="js" />
<label for>Javascript<label><br />
<input type="radio" name="langage" value="php" checked="checked"/>
<label for>php<label><br />
<input type="radio" name="langage" value="java"/>
<label for>java<label><br />
<input type="radio" name="langage" value="perl" disabled="disabled"/>
<label for>Perl<label><br />
<input type="reset" value="Annuler" size="40"/>
<input type="reset" value="Annuler (d&#xe9;sactiv&#xe9;)" disabled="disabled"/>
</form>
<p>Si vous cliquez le premier bouton "Annuler", le formulaire va etre initialisé.
Remarquez que l'autre bouton est d&#xe9;sactiv&#xe9;</p>
</body>
</html>
```

checkbox.html | view-source:file:///C:/Us...

file:///C:/Users/Neila/checkbox.html

Javascript

php

java

Perl

Annuler | Annuler (désactivé)

Si vous cliquez le premier bouton "Annuler", le formulaire va etre initialisé. Remarquez que l'autre bouton est désactivé

<input type = “button”>



- Il N'a de sens que dans un contexte JavaScript
  - Pas de comportement préprogrammé
  - Ne permet pas de collecter une valeur

Syntaxe:

```
<input type="button" name="nom" value="bouton"/>
```

- Attributs :
  - Name, value, size, disabled="disabled"

Exemple :

```
<form>…
<input type="button" value="Calculez" name="somme" size="40"/>
<input type="button" value="Multipliez(désactivé)"
name="produit" disabled="disabled"/>
</form>
```


neila.benlakhal@gmail.com

<input type = “button”> : example




```
<html>
<body>
<form name="input" action="html_form_action.php">
<label for="nombre">Nombre:</label><input type="text" size="30" />
<input type="button" value="Calculez" name="somme" size="40"/><br/>
<label for="carre">Le carr&#xe9; du nombre est :</label>
<input type="text" size="30" /><br/>
<input type="button" value="Multipliez(d&#xe9;sactiv&#xe9;)" name="produit" disabled="disabled"/>
</form>
```

checkbox.html | view-source:file:///C:/Us...

file:///C:/Users/Neila/checkbox.html

Nombre: [ ] Calculez

Le carré du nombre est :

[ ]

Multipliez(désactivé)

Zone de texte libre



- Zone de saisie de texte libre qui permet de saisir du texte sur plusieurs lignes et colonnes

Syntaxe :

```
<textarea name="nom" rows=".." cols= "..">
Contenu qui sera affiché par défaut
</textarea>
```

- Attributs :
  - name, readonly, disabled="disabled"
  - **rows,cols**

Exemple :

```
<textarea rows="10" cols="20" name="ftexte" >
At W3Schools you will find all the Web-building tutorials you
need, from basic HTML to advanced XML, SQL, ASP, and PHP.
</textarea>
```

Zone de texte libre : exemple




```
<html>
<body>
<form name="input" action="html_form_action.php">
<h3> Information: </h3>
<textarea rows="10" cols="20" name="ftexte" >
At W3Schools you will find all the Web-building tutorials you need, from basic HTML to
advanced XML, SQL, ASP, and PHP.
</textarea>
```

checkbox.html | view-source:file:///C:/Us...

file:///C:/Users/Neila/checkbox.html

Information:

At W3Schools you will find all the Web-building tutorials you need, from basic HTML to advanced XML, SQL, ASP, and PHP.

neila.benlakhal@gmail.com

Zone de sélection/liste déroulante



- La balise **‹select›** permet de définir une liste de déroulante (drop-down list). Chaque choix dans la liste est défini par une balise **‹option›**

Syntaxe :

```
<select>
  <option value="value1">Valeur 1</option>
  <option value="value2">Valeur 2</option>  …
</select>
```

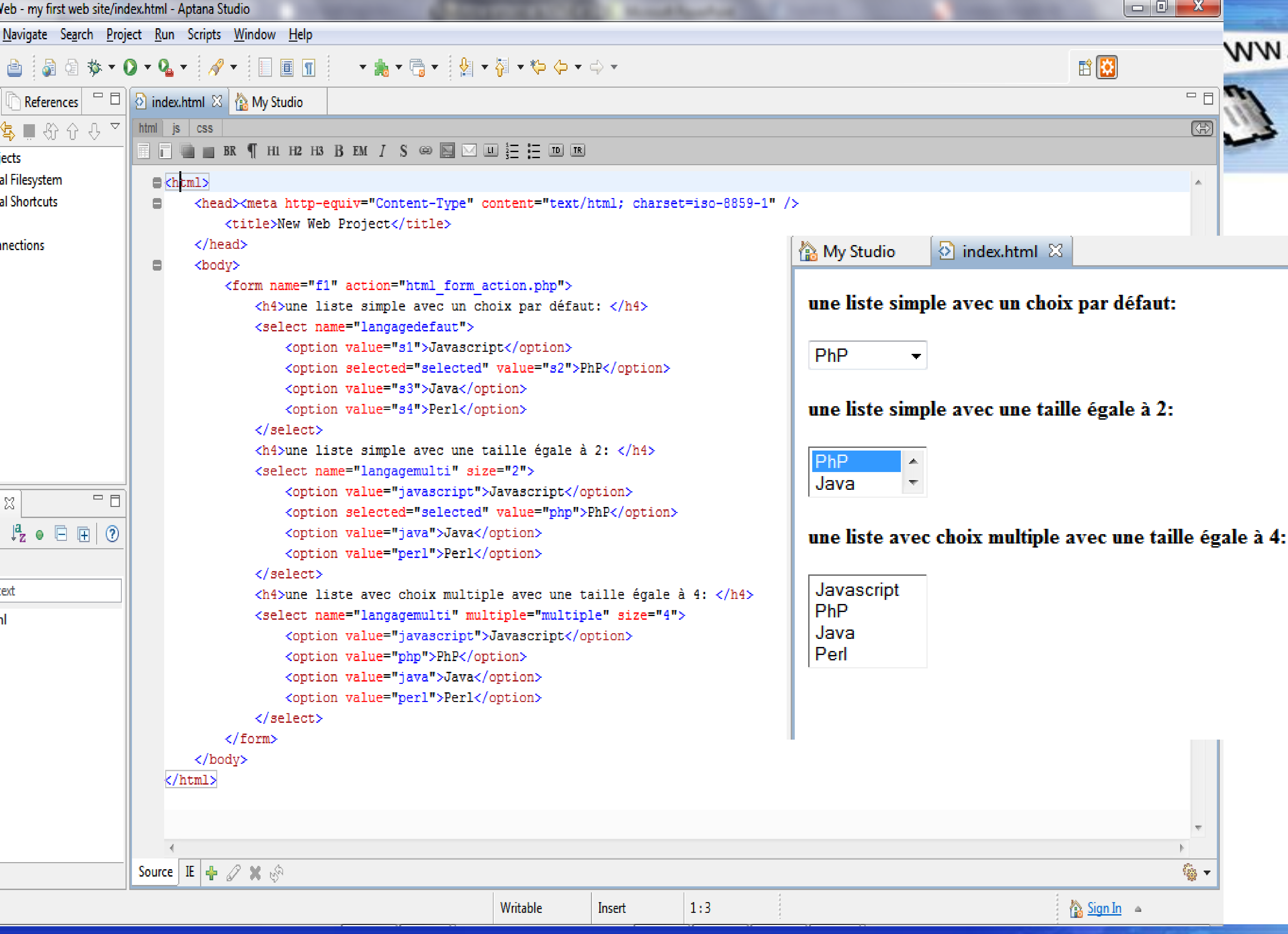
- Attributs de **‹select›**
  - **Size**: spécifie le nombre d'option visible dans la liste
  - **Multiple**: spécifie que plusieurs options peuvent être sélectionnées à la fois
  - **Name, disabled**

- Attributs de **‹option›**
  - **Value**: spécifie la valeur à envoyer au serveur si cette option est sélectionnée
  - **Selected**: spécifie que cette option est sélectionnée par défaut
  - **Label**: défini un label
  - **Disabled**

Zone de sélection : exemple




```
…
        <form name="f1" action="html_form_action.php">
            <h4>une liste simple avec un choix par défaut: </h4>
            <select name="langagedefaut">
                <option value="s1">Javascript</option>
                <option selected="selected" value="s2">PhP</option>
                <option value="s3">Java</option>
                <option value="s4">Perl</option>
            </select>
        <h4>une liste avec choix multiple avec une taille égale à 4: </h4>
            <select name="langagemulti" multiple="multiple" size="4">
                <option value="javascript">Javascript</option>
                <option value="php">PhP</option>
                <option value="java">Java</option>
                <option value="perl">Perl</option>
            </select>
        </form>
```


Web - my first web site/index.html - Aptana Studio
Navigate Search Project Run Scripts Window Help
References
index.html
My Studio
html js css
<html>
<head><meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>New Web Project</title>
</head>
<body>
<form name="f1" action="html_form_action.php">
<h4>une liste simple avec un choix par défaut: </h4>
<select name="langagedefaut">
<option value="s1">Javascript</option>
<option selected="selected" value="s2">PhP</option>
<option value="s3">Java</option>
<option value="s4">Perl</option>
</select>
<h4>une liste simple avec une taille égale à 2: </h4>
<select name="langagemulti" size="2">
<option value="javascript">Javascript</option>
<option selected="selected" value="php">PhP</option>
<option value="java">Java</option>
<option value="perl">Perl</option>
</select>
<h4>une liste avec choix multiple avec une taille égale à 4: </h4>
<select name="langagemulti" multiple="multiple" size="4">
<option value="javascript">Javascript</option>
<option value="php">PhP</option>
<option value="java">Java</option>
<option value="perl">Perl</option>
</select>
</form>
</body>
</html>
une liste simple avec un choix par défaut:
PhP
une liste simple avec une taille égale à 2:
PhP
Java
une liste avec choix multiple avec une taille égale à 4:
Javascript
PhP
Java
Perl
Source IE
Writable Insert 1:3
Sign In



Incorporer des objets MultiMedia en HTML 4.01

Types d'objets



- Par objet, on entend tout type de fichier situé hors d'un fichier HTML et devant y être incorporé:
  - Un fichier de données (un tableau Excel)
  - Un dessin AUTO CAD,
  - Un fichier de musique Midi,
  - Une animation Flash,
  - Une source en transit (streaming) pour la transmission radio,etc.
  - Mais il peut s'agir aussi d'un fichier exécutable par le navigateur Web, à savoir d'un programme. Ex: des applets Java ou des contrôles ActiveX.
- Pour tous les multimédia et références d'autres programmes: le repère `<object>…</object>` est utilisé.

Types d'objets



- Ce repère ne peut certes pas résoudre le problème de l'affichage d'un fichier quelconque chez l'utilisateur mais il propose tout au moins une syntaxe uniforme et contribue de ce fait à la simplification de HTML.
- On spécifie de quelle sorte d'objet il s'agit avec l'attribut **type** en indiquant le type mime, et la source du document avec l'attribut **data** en indiquant son URI.
- Vous devez toujours noter les mentions de largeur (`width=`) et de hauteur (`height=`) .

Syntaxe:

```
<object data="…" type="…" width="…" height="…">
</object>
```


neila.benlakhal@gmail.com

Contenu de la balise <object>



- Pour des fichiers de données référencés avec data=, notez en plus l'attribut type=, c'est le type Mime du fichier.
- **Mime** pour **M**ultipurpose **I**nternet **M**ail **E**xtensions.
  - Conçu à l'origine pour les systèmes de messagerie (E-mails) comportant divers type de fichiers attachés; Il fallait trouver une convention à l'intérieur du fichier pour en différencier les différentes parties (par exemple le texte du courriel et le fichier ZIP joint).
  - Après, il s'est avéré utiles pour différents types de communication client/serveurs.
  - Un type Mime comprend 2 parties: la mention d'un type de média et la mention d'un sous-type, séparées par une barre oblique.
  - Exemples: text/html, image/gif, text/css.
  - liste complète : http://www.w3schools.com/media/media_mimeref.asp

Exemples



Exemples :

- incorporer un document HTML dans un autre document:

```
<object data="data/test.html" type="text/html"
width="300" height="200">
</object>
```

- incorporer un document pdf

```
<object data="data/test.pdf" type="application/pdf"
width="300" height="200">
alt : <a href="data/test.pdf">test.pdf</a> </object>
```


neila.benlakhal@gmail.com

Autres exemples



- Vous avez parfois besoin de spécifier des paramètres relatifs au document par l'intermédiaire de la balise **<param>** :

Exemple : incorporer un document wav

```
<object type="audio/x-wav" data="data/test.wav"
width="200" height="20">
 <param name="src" value="data/test.wav">
 <param name="autoplay" value="false">
 <param name="autoStart" value="0">
 alt : <a href="data/test.wav">test.wav</a>
 </object>
```

- *Le paramètre autoplay est compréhensible par QuickTime, autoStart par Windows Media Player et RealAudio.*



La balise META
Les types de document HTML 4.01

neila.benlakhal@gmail.com

Les balises meta 1/3



- Les balises **<meta>** placées dans le repère **<head>** peuvent avoir plusieurs rôles:
  - identifier les propriétés d'un document (par exemple, l'auteur, la date d'expiration, une liste de mots-clés, etc.) et assigner des valeurs à ces propriétés :
  - Exemples :

```
<meta name="Auteur" content="Neila BL">
<meta name="copyright" content="&copy; 2010 NBL inc.">
<meta name="mots-cles" content="emploi,informatique">
<meta name="date" content="2012-11-06T08:49:37+00:00">
```

Les balises meta 2/3



- spécifier des mots-clés qu'un moteur de recherche peut utiliser pour améliorer la pertinence du résultat d'une recherche.
  - Quand des éléments META fournissent des informations en plusieurs langues sur un document, les moteurs de recherche peuvent opérer un filtrage, en fonction de l'attribut **lang** :
- Par exemple :

```
<!-- Pour les américanophones -->
<META name="keywords" lang="en-us"
         content="vacation, Greece, sunshine">
<!-- Pour les anglophones-->
<META name="keywords" lang="en"
         content="holiday, Greece, sunshine">
<!-- Pour les francophones -->
<META name="keywords" lang="fr"
         content="vacances, Gr&egrave;ce, soleil">
```

Autres utilisation de <meta>



- On peut utiliser l'élément META pour spécifier les informations par défaut sur un document dans les cas suivants :
  - le langage de script par défaut ;
  - le langage de feuille de style par défaut ;
  - l'encodage de caractères du document.

Mention du jeu de caractères par défaut



- Vous pouvez à l'aide d'un repère **‹meta›** décider **explicitement** quel jeu de caractères vous utilisez dans le fichier HTML.

Exemple:

```
<head>
<meta http-equiv="content-type" content="text/html;
charset=ISO-8859-1">
<!-- ... autres mentions de l'entête de fichier ... -->
</head>
```

- La mention du jeu de caractères est définie grâce à http-equiv="content-type".
- iso-8559-1 : c'est le jeu de caractères normal pour les langues d'Europe de l'ouest, parmi les quelles figure également le Français.

Mentions pour les langages par défaut pour les scripts et feuilles de style



- Pour les langages complémentaires à HTML comme les feuilles de style et les scripts, vous pouvez déterminer les langages que vous utilisez dans le fichier HTML.

```
<head>
<meta http-equiv="Content-Script-Type" content="text/javascript">
<meta http-equiv="Content-Style-Type" content="text/css">
<!-- ... autres mentions de l'entête de fichier ... -->
</head>
```


neila.benlakhal@gmail.com

Mentions du type de document



- Les règles syntaxique pour HTML sont formulées à l'aide de SGML, les règles pour XHTML à l'aide de XML.
- **D'après les règles des langages de marquage, un fichier HTML n'est valide que s'il mentionne un certain type de document pour s'en tenir ensuite dans le reste du code-source aux règles qui sont définies pour ce type de document.**
- Derrière chaque mention de type de document il y a ce qu'on appelle des définitions de type de document (DTD).
- Dans une DTD y sont fixés les éléments que peut contenir un document HTML, quels éléments peuvent figurer dans quels ordres, quels attributs s'appliquent à un élément, si la mention de ces attributs est facultative ou obligatoire etc.
- Exemple d'une mention de type de document:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01
Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
```

Explication DOCTYPE




```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/strict.dtd">
```

- DOCTYPE HTML PUBLIC: Cela signifie, que vous vous référez au DTD HTML disponible publiquement. W3C est l'éditeur du DTD.
- DTD HTML 4.01 Transitional signifie que vous utilisez le type de document "HTML" et cela dans la version du langage 4.01 et sa variante "Transitional".
- Le terme EN est une abréviation pour la langue qui veut dire ici l'anglais (la langue parlée dans laquelle les noms d'éléments et d'attributs du langage de repères ont été définis)
- Par l'adresse Web mentionnée, un logiciel de lecture peut appeler la définition du type de document (DTD) pour y "consulter" les règles qui y sont notées.

 neila.benlakhal@gmail.com

Variantes DOCTYPE



- **Mentions correspondantes de type de document pour XHTML:**

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
```

DE HTML4.01 à HTML5



- L'entête d'un document HTML4.01 omis dans tout les exemples du cours:

```
<?xml version="1.0"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="en" xml:lang="en">
<head>
<meta name="generator" content="HTML Tidy, see www.w3.org" />
<title>XHTML 1.0: The Extensible HyperText Markup Language (Second
Edition)</title>
<link rel="stylesheet" type="text/css" media="screen" href="xhtml.css" />
<link rel="stylesheet" type="text/css" media="screen"
href="http://www.w3.org/StyleSheets/TR/W3C-REC.css" />
</head>
<body>…

</body>
</html>
```

Problèmes avec HTML4.01



- Une multitudes de versions
  - HTML4, HTML4,01, XHTML2,0, XHTML1,0, etc..
  - La version strict, transationnal, frameset, etc.
- Une multitude de variante de la balise **Meta** en fonction du jeu de caractère
- Complexité de la définition du **Doctype**
- HTML5 est là!
  - Première contribution: simplification du DOCTYPE et META pour garantir une compatibilité exhaustive avec tout type de navigateur quelque soit sa version !

Les balises principales



- Le **doctype** est simplifié :

Syntaxe : <!DOCTYPE html>

- Il n'est pas sensible à la casse (on peut écrire <!doctype html> par exemple).
- Avant le Doctype pour un document xhtml 1.0 s'écrivait:

```
<!DOCTYPE html
          PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
          "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
strict.dtd">
```

```html
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
	<title>Head First Lounge</title>
	<meta http-equiv="content-type" content="text/html;
charset=UTF-8">
</head>
<body>
	<h1>Welcome to Head First Lounge</h1>
	<p>
		<img src="drinks.gif" alt="Drinks">
	</p>
	<p>
Join us any evening for refreshing <a
href="elixirs.html">elixirs</a>,
conversation and maybe a game or two of Ta
Revolution.
Wireless access is always provided; BYOWS (I
Web Server).
	</p>
```

```html
<!doctype html>
<html>
<head>
	<title>Head First Lounge</title>
	<meta charset="utf-8">
	<link rel="stylesheet" href="lounge.css">
	<script src="lounge.js"></script>
</head>
<body>
	<h1>Welcome to Head First Lounge</h1>
	<p>
		<img src="drinks.gif" alt="Drinks">
	</p>
	<p>
Join us any evening for refreshing <a href="elixirs.html">elixirs</a>,
conversation and maybe a game or two of Tap Tap Revolution.
Wireless access is always provided; BYOWS (Bring Your Own Web Server).
	</p>
</body>
</html>
```



HTML5



Comme son nom l'indique, HTML 5 est le successeur de HTML 4.01.

Le travail sur HTML5 a commencé fin 2003 grâce à un groupe de travail indépendant : **WHATWG(Web Hypertext Application Technology Working Group)** en se basant principalement sur Web Forms 2.0.

Depuis 2006, W3C intègre ce projet est travaille en étroite collaboration avec WHATWG.

De là s'est fait un gros travail afin de permettre à HTML5 d'être rétro-compatible avec ses ancêtres, ce qui a quelque peu ralenti son développement.

Date estimative de sa sortie en standard en 2014

(as of December 2013, is a W3C Candidate Recommendation.)



... *HTML5 does not belong to a company or a specific browser. It has been forged by a community of people interested in evolving the web and a consortium of technological leaders that includes Google, Microsoft, Apple, Mozilla, Facebook, IBM, HP, Adobe, and many others. The community and consortium continue to collaborate on universal browser standards to push web capabilities even further. The next generation of web apps can run high-performance graphics, work offline, store a large amount of data on the client, perform calculations fast, and take interactivity and collaboration to the next level.* ...

http://www.html5rocks.com/en/

Les buts de HTML5



- Minimiser le recours à des technologies/plugins tel que **Adobe Flash**, **JavaFX**, and **Microsoft Silverlight**
- Concrétiser la vision **Rich internet Application (RIA)** sans avoir recours à des technologies tierces ou à la partie serveur :
  - Des application aussi riches en fonctionnalités que les « desktop application » qui tournent sur toute versions de navigateurs et qui ne requiert pas l'installation d'un plugin et ne fait appelle à la partie serveur que quand il s'agit d'un accès au données.

RIA ?



- A rich Internet application (RIA) is a Web application designed to deliver the same features and functions normally associated with deskop applications. RIAs generally split the processing across the Internet/network divide by locating the user interface and related activity and capability on the client side, and the data manipulation and operation on the application server side.
- An RIA normally runs inside a Web browser and usually does not require software installation on the client side to work. However, some RIAs may only work properly with one or more specific browsers.

(http://searchsoa.techtarget.com/definition/Rich-Internet-Application-RIA)

Promesses de HTML5



- Usage extensif de Javascript (et ses différentes librairies intégré dans HTML5) et de AJAX pour se rapprocher des desktop applications en terme de fonctionnalités.
- Ajout de nouvelles balises ayant une sémantique
  - Indexation beaucoup plus facile des sites par les moteurs de recherche (Google).
  - Recherche plus efficace.
- En HTML4.01, tout est de type string, des contrôles excessifs des champs avec des scripts
  - Introduire de nouveau types (Date, etc…)
- Avec HTML4,01, toutes les données sont stockées coté serveurs
  - Se réserver un espace de stockage coté client.

Promesses de HTML5



- Parfaite séparation entre le contenu et la forme
  - En séparant plusieurs éléments comme : ‹big›, ‹font›, ‹strike›, ‹u›, ‹center›, …
- Haut niveau d'interopérabilité
  - Même comportement peu importe le navigateur utilisé (avant l'API DOM différait d'un navigateur à un autre)
- Accès universel au document de divers périphériques, de diverses plateformes
- Plus de simplicité (simplification de Meta, du Doctype, etc…)
- Éviter les plugin externes vu que :
  - Ils peuvent être bloquer
  - Certains s'intègrent mal dans HTML
  - Ils se plantent facilement,

Balises supprimées

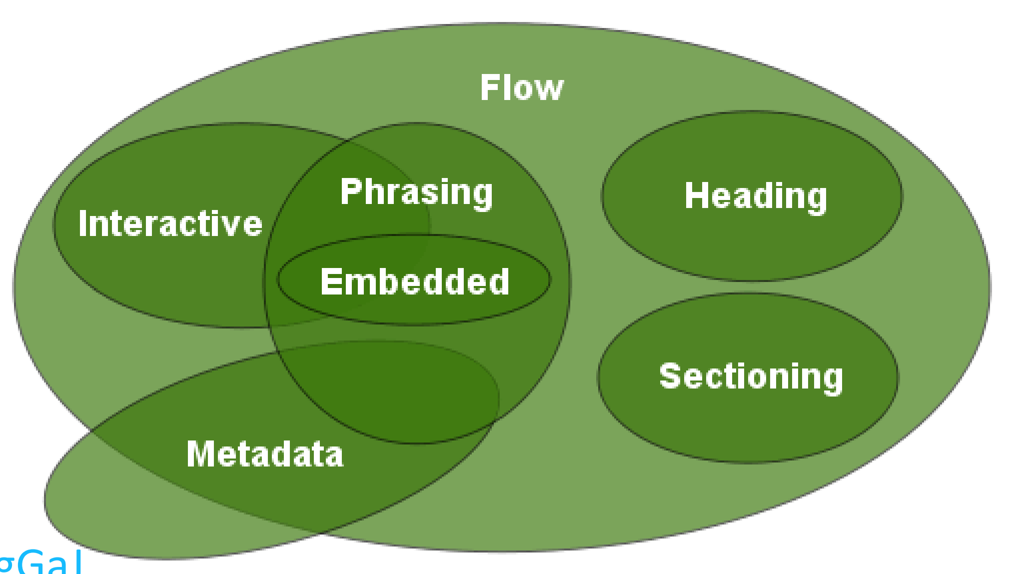


- Ces balises figuraient dans HTML4.01 mais plus dans HTML5
  - <acronym>
  - <applet>
  - <basefont>
  - <big>
  - <center>
  - <dir>
  - <font>
  - <frame>
  - <frameset>
  - <noframes>
  - <strike>
  - <tt>

La nouveauté dans HTML5



- Une grande nouveauté annoncée au niveau structurel des éléments est signalée dans l'introduction : **The new content model**
- Un nouveau schéma : les éléments HTML sont à présent uniquement regroupés en catégories, sachant que les éléments peuvent apparaître dans plusieurs catégories :
  - Metadata content
  - Flow content
  - Sectioning content
  - Heading content
  - Phrasing content
  - Embedded content
  - Interactive content



http://www.youtube.com/watch?v=YFuzqgGaJPQ

Avant ce modèle comment été structuré HTML4.01 ?



- **Block content**
  - Les éléments qui occuperaient une ligne dans un document HTML
  - Exp : <p>, <div>, …
- Inline content
  - les éléments qui se trouveraient dans des éléments de bloc
  - Exp : <a>, <span>, …

  - Ce modèle permettait seulement de structurer le document. Aucun sens n'est octroyé au différentes parties.

neila.benlakhal@gmail.com

HTML5 content model



http://www.youtube.com/watch?v=YFuzqgGaJPQ

- **Embedded content** : any content that imports other resources into the document. Exp. : <object>, <video>, <canvas>, <embed>, etc.
- **Interactive content** : any content specifically intended for user interaction.
  - Exp; : <details>, <object>, <video>, etc.
- **Heading content** : defines the header of a section, which can either be explicitly marked up with sectioning elements or implied by the heading content itself. Exp: <h1> à <h6.
- **Phrasing content**: is the text of the document as well as elements used markup the text within paragraph level structures (inline content from the HTML 4 specification).
- **Sectioning content** is content that defines the scope of headings and footers. Using these elements will create a new section within the document.
  - Exp: <article>, <aside>, <nav>, <section>, etc.
- **Flow content :** contains the majority of elements in HTML5.Think of these elements as elements that would be included in the normal flow of the document.
- **Metadata content** is defined as being content that sets up the presentation or behavior of the rest of the content. You'll primarily find these elements in the head of the document. Exp: <link>, <meta>, <script>, <title>, etc.

neila.benlakhal@gmail.com

Comment HTML5 va achever ses promesses ?



- En se basant sut HTML5 APIs + un ensembles de technologies connexes:

HTML5

Taxonomy & Status (December 2011)

- W3C Recommendation
- Candidate Recommendation
- Last Call
- Working Draft
- Non-W3C Specifications
- Deprecated W3C APIs

HTML5 & related technologies

CSS3
MathML 3.0
SVG
Selectors L1
Navigation Timing
Web Open Font
XmlHTTP Request 1
Geo Location
Contacts API
WebGL
HTTP Caching
RDFa
WAI-ARIA
Device Orientation
Animation Timing
Calendar API
Touch Events
File API
Media Capture
Indexed Database

W3C HTML5 specification

Web SQL
Web Storage
HTML + RDFa

Initial WHATWG HTML5 specification

Web Messaging
Canvas 2D
Web Sockets
HTML Markup
Audio Video
Drag and Drop
Web Workers
Micro data

Nouvelles règles



- Dans HTML5, il n'est pas systématiquement nécessaire de fermer tous les éléments. Ainsi, les éléments <p>, <dd>, <dt>, <li>, etc. n'ont plus besoin de balise fermante pour être valides.
- **Seule la version XHTML 5 obligera à fermer ces éléments !!!**
- Certains éléments ne nécessiteront ni balise fermante ni balise ouvrante, c'est le cas de <html>, <head>, <body>, <thead>, <tfoot> et <tbody>.
- **Cela signifierait que la présence même de ces éléments deviendrait implicite!!!**

neila.benlakhal@gmail.com

Nouveautés de HTML5



- Nouveaux éléments sémantiques
- Nouveaux éléments et attributs pour les formulaires
- Video/audio natifs (sans le recours à un Plugin externe)
- Canvas drawing widget : outil de dessin
- Web storage
- Offline application
- Web workers
- Geolocalisation
- Web socket

neila.benlakhal@gmail.com

HTML5



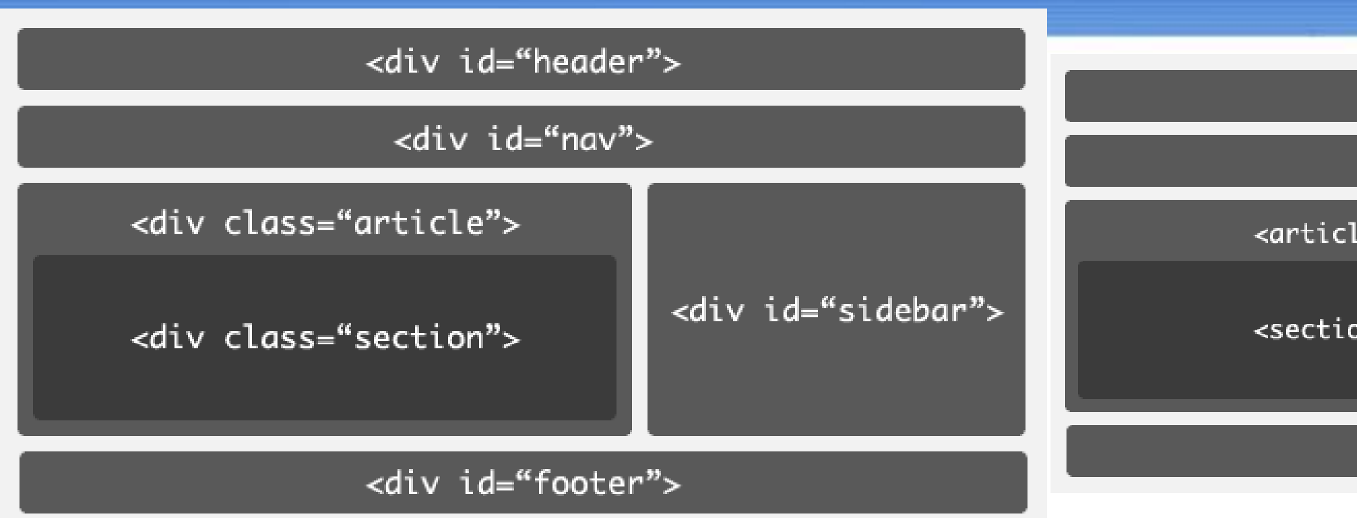
- Le HTML5 introduit une série complète de nouveaux éléments qui font qu'il est beaucoup plus simple de structurer les pages.

  - La plupart des pages HTML 4 contiennent une diversité de structures identiques, comme les en-têtes, les bas de page, les colonnes les barres de navigations etc.

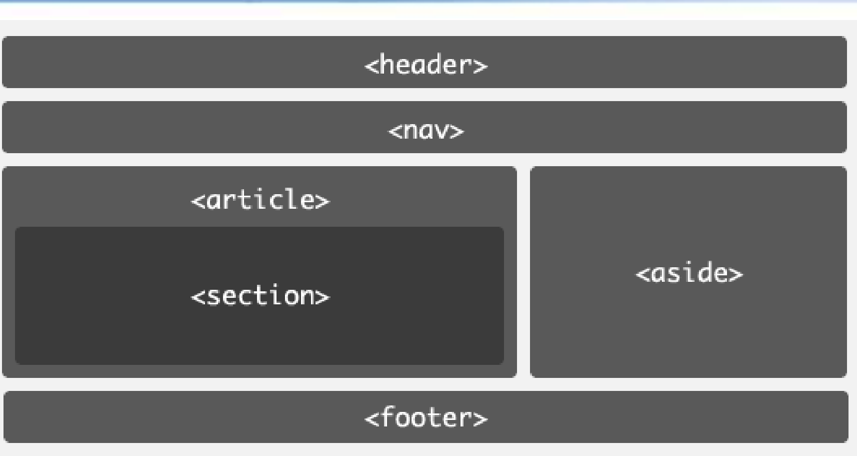
  - Aujourd'hui, il est relativement fréquent de les baliser par des éléments ***div***, leur attribuant à chacun un identifiant ou une classe descriptive pour pouvoir les styler plus tard.

De HTML4 à HTML5





HTML 4.01



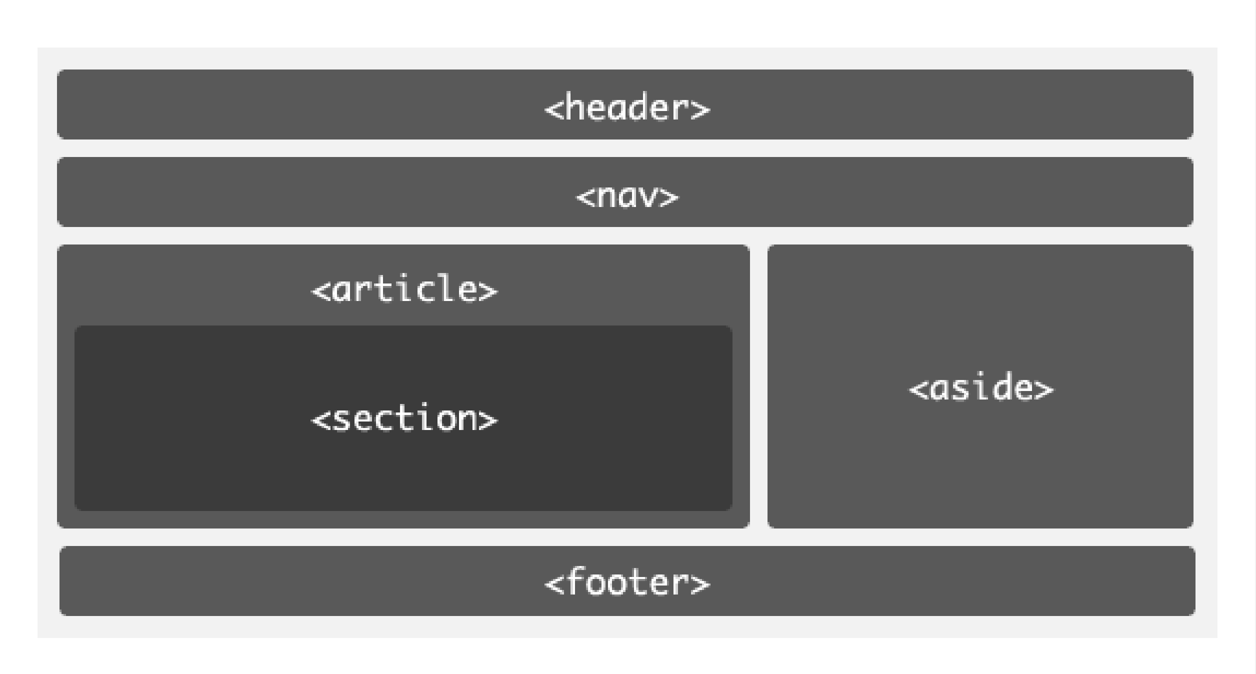
HTML5

- Ce diagramme illustre une présentation typique à 2 colonnes, balisée à l'aide de ***div*** ayant des attributs **id** et **class**. Elle contient un en-tête, un pied de page, et une barre de navigation horizontale sous l'en-tête. Le contenu principal comprend un article et une barre latérale à sa droite.
- L'utilisation d'éléments ***div*** est due principalement aux versions actuelles du HTML 4 qui n'ont pas la **sémantique** nécessaire pour décrire ces parties plus spécifiquement.
- Le HTML 5 s'attèle à ce problème en introduisant de nouveaux éléments pour représenter chacune de ces différentes sections.

Nouvelle structuration



- Les éléments div ont été remplacés par de nouveaux éléments : ***header***, ***nav***, ***section***, ***article***, ***aside*** et ***footer***.
- Le code de ce document pourrait ressembler à celui-ci :




```
<!DOCTYPE html>
<body>
<header>...</header>
<nav>...</nav>
<article>
  <section>...</section>
</article>
<aside>...</aside>
<footer>...</footer>
 </body>
```

```
<!doctype html>
<html>
<head>
    <title>Page title</title>
</head>
<body>
    <header>
        <h1>Page title</h1>
    </header>
    <nav>
        <!-- Navigation -->
    </nav>
    <section id="intro">
        <!-- Introduction -->
    </section>
    <section>
        <!-- Main content area -->
    </section>
    <aside>
        <!-- Sidebar -->
    </aside>
    <footer>
        <!-- Footer -->
    </footer>
  </body></html>
```

<header>

<nav>

<article>

<section>

<aside>

<footer>

L'élément *header*



- L'élément **‹header›** représente l'en-tête du document. Les en-têtes peuvent contenir plus qu'un simple titre de la page/un logo, un formulaire de recherche, un titre.
- Il peut être utiliser pour introduire la page comme pour introduire une autre section plus tard dans le document.
- Exemple :

```
<header>
    <h1>A Preview of HTML 5</h1>
</header>
<p>The rest of my home page...</p>
…
```


neila.benlakhal@gmail.com

<footer> et <nav>



- L'élément **<footer>** représente le bas de la section à laquelle il s'applique. Un *pied* contient typiquement une information sur sa section comme son auteur, des liens vers des documents liés, les données de copyright et autres données du même type.
  - Exemple :

```
<footer> copyright 2013 Neila Inc.</footer>
```

- L'élément **<nav>** représente une section de liens de navigation. Il convient à la fois pour la navigation dans le site ou une table des matières.
  - Exemple :

```
<nav>
    <a href="default.php">Home</a>
    <a href="tag_meter.php">Previous</a>
    <a href="tag_noscript.php">Next</a>
</nav>
```

<aside>



- L'élément **<aside>** est typiquement utile pour baliser des barres latérales.
- Il est destiné au contenu indirectement lié à l'article lui-même, il représente ce qui l'entoure comme par exemple une barre latérale d'archives.
- Exemples :

```
<aside>
    <h1>Archives</h1>
    <ul><li><a href="/archives/12/05/">
               Mai 2012</a></li>
        <li><a href="/archives /12/06/">
              Juin 2012</a></li>
        <li><a href="/archives /12/07/">
              Juillet 2012</a></li></ul>
</aside>
```

```
<!DOCTYPE HTML>
<body>
<p>Me and my family visited The
Epcot center this summer.</p>
<aside>
<h4>Epcot Center</h4>
The Epcot Center is a theme park
in Disney World, Florida.
</aside>
</body></html>
```

<article>



- L'élément **<article>** représente un texte comme par exemple un article de journal, de blog ou de forum.
- Exemple :

```
<article>
<p><a href="http://www.alsacreations.com/">
XHTML est mort, vive HTML5 !</a><br />
Sous ce titre quelque peu provocateur (et faux) se
cache une réalité officielle depuis hier soir : le W3C
vient d'annoncer que ses travaux sur HTML5 se
termineront en 2014.</p>
</article>
```


neila.benlakhal@gmail.com

L'élément <section>



- L'élément **<section>** permet de définir les différentes sections d'un document comme par exemple les chapitres, les en-tête et pied-de-page, ou toute autre section dans un document.
- Il peut être combiné avec les éléments h1, h2, h3, h4, h5, et h6 pour une meilleure définition de la structure du document.
- Exemple :

```
<article>
<header>
<h1>Welcome</h1>
</header>
<section>
<h4>What We Do</h4>
<p>We protect sharks...</p>
</section>
</article>
```

Pourquoi de telles balises?



- L'introduction des éléments **Header**, **nav**, **aside**, **footer**, **article** et **section** permet de remplacer l'utilisation de l'élément div

- **Les nouvelles balises ont un sens (sémantique)!!!**

- Avantage: Améliorer la recherche dans les pages et mieux évaluer le contenu d'une page.
- Concrétiser la vision Web 3.0.



Les formulaires HTML (5)

Nouveaux éléments et Attributs

HTML5 a enrichi les formulaire d'un nouveau ensemble de balises, de types et d'attributs pour ne plus avoir besoin de la validation de formulaire coté client par des scripts (de type JavaScript par exemple).

Nouveaux éléments comme : <datalist> ou encore **<output>** ont été inspiré du travail de web forms 2.0 (whatWG)

HTML5 : nouveaux éléments de formulaire



- La balise **‹datalist› :** Elle définit une liste d'options à utiliser avec l'élément **‹input›**, pour définir les valeurs permise pour cette zone de saisie.
  - ***Le datalist et ses options ne seront pas affichés!!***
- L'attribut **list** de **‹input›** permet de lier cet élément avec le **‹datalist›**.

Exemple :

```
<!DOCTYPE HTML><html><body>
My car : <input list="cars" />
<datalist id="cars">
      <option value="Fiesta">
      <option value="Ford">
      <option value="Focus">
</datalist></body></html>
```

My car : F

Fiesta
Ford
Focus

Output



- **‹output›** : est utilisé pour afficher le résultat d'un calcul effectué par un script JS par exemple.

```
<!DOCTYPE HTML>
<html><body>
<form name="calcul1" oninput="resultat1.value =
parseInt(valeur1.value) * parseInt(valeur2.value);">
  <input type="text" size="3" name="valeur1"
value="15" /> *
  <input type="text"  size="3" name="valeur2"
value="10" /> =
  <output for="valeur1 valeur2"
name="resultat1">150</output>
</form>
</body></html>
```

15 * 10 = 150

Les nouveaux attributs





HTML5 new types

Username (required):

Step number

Hint — First name

Three letter country code(Country code with regular expression) :

Select images: ファイル選択 選択されていません

Quantity (between 1 and 5)

Enter a date after 2000-01-01: 年/月/日 ▼

<< < 2012年(平成24年) 10月 ▼ > >>

日	月	火	水	木	金	土
30	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31	1	2	3
4	5	6	7	8	9	10

今日 クリア

Submit as normal

Submit to a new window

Nouveaux types de <input> et nouveaux attributs



Step number : <input type="**number**" name="points" **step**="3">

Username: <input type="text" name="usrname" **required="required**“ **autofocus**>

Hint:<input type="text" name="fname" **placeholder**="First name">

Country code with regular expression :
<input type="text" name="country_code" **pattern**="[A-Za-z]{3}" title=“3 letter country code">

Select images: <input type="file" name="img" **multiple="multiple"**>

Enter a date after 2000-01-01:<input type="**date**" name="bday" min="2000-01-02">
Quantity (between 1 and 5):
<input type="**number**" name="quantity" **min**="1" **max**="5">
submit button as an image:
<input type="**image**" **src**="img_submit.gif" alt="Submit" width="48" height="48">
<input type="submit" value="Submit as normal">
<input type="submit" **formtarget="_blank"** value="Submit to a new window">

```
<!DOCTYPE HTML>
<html>
    <head>
        <title>HTML5 Tags</title>
    </head>
    <body>
        <form action="index.php" method="post" name="f1" id="f1">
            E-mail: <input type="email" name="user_email" /><br/>
            Range : <input type="range" name="points" min="1" max="10" /><br/>
            Points: <input type="number" name="points" min="1" max="10" /><br/>
            Homepage: <input type="url" name="user_url" /><br/>
            Date: <input type="date" name="user_date" /><br/>
            Date and time: <input type="datetime-local" name="user_date" /><br/>
        </form>
    </body>
</html>
```

DEMO

HTML5 Tags

127.0.0.1:8000/my%20first%20web%20site/HTML5.html

E-mail: neila

Range :

Points: 2

Homepage:

Date: 1582-10-18

Date and time: 1582-10-15T00:04

HTML 5: les nouveaux attributs pour formulaire et ses champs



- Les nouveaux attributs pour **<form>** :
  - Autocomplete : le champ du formulaire doit avoir une fonction de saisie semi-automatique.

```
E-mail: <input type="email" name="email" autocomplete="off" />
```

- Les nouveaux attributs pour **<input>** :
  - Autofocus : spécifie qu'un champ doit automatiquement avoir le focus lorsque la page est chargée

```
User name:<input type="text" name="user"  autofocus="autofocus" />
```

  - Form : indique a quel formulaire(s) le champs input appartient

neila.benlakhal@gmail.com

<video>



- L'élément **‹video›** permet d'insérer des vidéos dans une page Web de façon standard SANS avoir recours à un plugin tel que ADOBE Flash ou Apple QuickTIME etc.
- Défini les attributs : **preload**, **autoplay**, **loop**, **src**, **controls**, pour indiquer comment la vidéo doit être lu:
- Les formats supportés:
  - WebM,MPEG4,Ogg/Theora video format

```
<!DOCTYPE HTML>
<html>
<body>
<video src="movie.ogg"  width="320" height="240" controls="controls">
If you see this text, Your browser does not support the video tag.
</video>
</body></html>
```

Les attributs de l'élément <video>



- L'attribut controls est un attribut booléen qui indique si l'auteur souhaite que cette interface utilisateur soit présente ou non par défaut.
- L'attribut facultatif poster peut être utilisé pour spécifier une image qui sera affichée à la place de la vidéo avant qu'elle ne commence.

```
<video src="video.ogg" controls="controls"
poster="poster.jpg"
width="320"
height="240">
 <a href="video.ogv">Télécharger le film</a>
</video>
```


neila.benlakhal@gmail.com

Exemple : <video>



Edit and Click Me >>

```
<!DOCTYPE HTML>
<html>
<body>

<video src="movie.ogg"  width="320" height="240"
controls="controls">
Your browser does not support the video tag.
</video>

</body>
</html>
```

Your Result:

00:12

Edit the code above and click to see the result

W3Schools.com - Try it yourself

neila.benlakhal@gmail.com

Video



- Les limitations :
  - Ne permet pas encore le respect des copyright/droit
  - Supporte mal l'utilisation des streaming, entrée micro/video des Webcam.
  - Beaucoup de travail à faire pour écarter les plugin tel ques adobe Flash ou encore apple Quicktime.

<audio>



- Il est aussi simple d'intégrer de l'audio à une page en utilisant l'élément **‹audio›**.
- L'élément **audio** n'a pas d'attributs **width**, **height** et **poster**.
- Exemple :

```
<audio src="/music/lostmojo.wav">
<p>If you are reading this, it is because your browser
does not support the audio element.</p>
</audio>
```


neila.benlakhal@gmail.com

<source>



- L'élément **‹source›** s'utilise avec les éléments ‹audio› et ‹video› pour indiquer des sources alternatives d'un fichier audio/video et entre lesquels le navigateur choisira en fonction des types/codec qu'il supporte.
- Lorsque l'élément *source* est utilisé, l'attribut ***src*** doit être omis de l'élément parent vidéo/audio.
- Exemple :

```
<audio>
      <source src="/music/good.wma" type="audio/x-ms-wma">
      <source src="/music/good.mp3" type="audio/mpeg">
      <p>If you are reading this, it is because your
browser does not support the HTML 'audio' element.</p>
</audio>
```


neila.benlakhal@gmail.com

Exemple <audio> :




```
<audio>
  <!-- Deux formats disponibles par ordre de priorité: -->
  <source src="trappeur.ogg" type="audio/ogg">
  <source src="trappeur.aac" type="audio/aac">
  <!-- Contenu alternatif si élément audio ou formats non
  supportés dans le navigateur: -->
  <a href="trappeur.ogg">
  Télécharger <cite>Avant j'étais trappeur</cite>
  </a>
  de David TMX (format Ogg Vorbis)
</audio>
```

Canvas 2D



- **Canvas›** : représente une zone de dessin pouvant afficher des graphiques
  - Elle permet une interactivité avec les utilisateurs voir même créer des jeux, choses qui nécessitaient auparavant l'utilisation d'autres technologies comme les applets ou les animations Flash etc.
  - l'API (faisant partie de la spécification de HTML5) qui lui est associée met à disposition du programmeur de nombreuses méthodes accessibles en JavaScript pour la création de forme et d'effets.
  - Cette API est utilisable en JavaScript : une fois la balise ‹canvas› créée (en spécifiant un identifiant), on instancie un contexte associé à cet élément sur lequel on effectuera les appels à l'API.

Canvas 2D



- On commence par créer un ‹canvas› et on lui associe un id
- On appelle cet id à partir de Javascript pour y dessiner des lignes, des formes etc…

Edit and Click Me »

```
<!DOCTYPE html>
<html>
<body>
<canvas id="myCanvas" width="200" height="100" style="border:1px
solid #c3c3c3;">
Your browser does not support the HTML5 canvas tag.
</canvas>
<script>
var c=document.getElementById("myCanvas");
var ctx=c.getContext("2d");
ctx.fillStyle="#FF0000";
ctx.fillRect(0,0,150,75);
</script>
</body>
</html>
```

Your Result:



Exemple avec Canvas



- http://html5demos.com/video-canvas

<figure>



- L'élément **‹figure›** peut être utilisé pour regrouper des éléments tels que des images ou des vidéos avec leur légende **‹figcaption›**.
- Exemples :

```
<figure>
<img src="/images/chaton.jpg" alt="oh le beau chaton" />
  <figcaption>Un petit chat mignon tout plein</figcaption>
</figure>
```

```
<p><a href="#1">Figure 1</a> provides the JavaScript code for creating
an alert box:</p>
<figure id="1">
<figcaption>Figure 1. JavaScript Alert Box.</figcaption>
<pre><code>alert('Hello!');</code></pre>
</figure>
```


neila.benlakhal@gmail.com

Canvas vs SVG



- SVG c'est quoi ??
  - SVG(scalable vector graphic) : permet de dessiner des graphiques vectorielles
  - Des graphiques qui d'adaptent à la taille de la fenêtre et qui ne sont pas affectés par le changement de résolution
  - Défini en XML imbriqué dans du HTML
  - Peuvent être contrôler via l'API SVG DOM

```
<!DOCTYPE html>
<html lang="en">
<head>...</head>
<body>
<svg xmlns="http://www.w3.org/2000/svg">
<circle id="myCircle" cx="50" cy="50" r="100" fill="blue" />
</svg>
</body>
</html>
```

Local storage



- Les serveurs utilisent le mécanisme de session/cookies pour reconnaitre les utilisateurs revenant à leurs sites
  - Les informations sont stockées en tant que cookies sur la machine de l'utilisateur
- Problèmes avec les cookies
  - Leurs tailles est limité (4k max/cookie)
  - Manque de fiabilité : elles peuvent être effacées par l'utilisateur
  - Doivent être ré-envoyer au serveur à chaque accès pour que le serveur reconnait l'utilisateur accédant à des pages différentes
- Plusieurs propositions de chez Adobe (Flash6), Microsoft IE, Google (Gears), etc. pour pouvoir stocker une plus grande quantité d'information sur la machine cliente
- HTML5 local storage permet de stocker un ensemble de paire clé/valeur
- La capacité de stockage n'est pas limitée.

neila.benlakhal@gmail.com

Local storage …



Exemple :

```
<!DOCTYPE html>
<html>
<body>

<div id="result"></div><br/>
<div id="resultmod"></div>
<script>
  localStorage.lecture="Web";
  document.getElementById("result").innerHTML="Lecture: " +
localStorage.lecture;
  localStorage.setItem('lecture','Web 3.0');
var  x= localStorage.getItem('lecture');
  document.getElementById("resultmod").innerHTML="Lecture modified : " + x;
 </script>
</body></html>
```

Session Storage



- sessionStorage vs. localStorage ??

- L'objet sessionStorage a le même but que l'objet localStorage, à une exception prêt:
  - sessionStorage garde les données le long d'une session du navigateur seulement,
  - localStorage garde les données indéfiniment jusqu'à ce que l'utilisateur les effacent en appelant la fonction clear().

neila.benlakhal@gmail.com

Web Workers



- Il arrive souvent que le navigateur se plante car un certain script tourne en arrière plan
- Les Web workers ont été proposé pour permettre que des scripts JS tournent en arrière plan sans bloquer le navigateur pendant que vous puissiez naviguer dans la fenêtre en toute liberté
- http://html5demos.com/worker
- http://www.w3schools.com/html/tryit.asp?filename=tryhtml5_webworker

Offline web applications



- HTML5 application cache permet d'accéder à une application Web même en mode Offline
- Ceci aura pour avantage :
  - Navigation optimisée vu que seulement le contenu qui a été sujet à des MAJ sera chargé à partir du serveur
  - Navigation plus rapide vu que les ressources stockées en local sont chargé plus rapidement
  - Navigation en mode Offline.

Geolocation



- HTML5 dispose d'une API qui permet la géolocalisation du navigateur du client moyennant une fonction JS **getLocation()** qui renvoie la position actuelle (longitude+latitude)
- http://www.w3schools.com/html/tryit.asp?filename=tryhtml5_geolocation

- http://html5demos.com/geo

- L'API de géolocalisation du W3C n'est pas supporté par IE.
- Prévoir l'utilisation d'une autre API tel que GEARS de Google, compatible avec tout OS /Navigateur.
- Les Smartphones utilisent leurs propres systèmes de géolocalisation différent de l'API intégrée de HTML5.

neila.benlakhal@gmail.com

Enfin



- http://caniuse.com/#cats=HTML5
  - Ce site indique les éléments HTML5 supportées par les différentes version des navigateurs à ce jour
- http://html5demos.com/
  - Un ensemble de DEMO des nouvelles API de HTML5
- http://refcardz.dzone.com/refcardz/html5-new-standards-web-interactivity
  - Un résumé des différentes API et leurs fonctions(.pdf)
- HTML5 n'est pas encore un standard !
- On se rapproche de plus en plus en terme de fonctionnalités de la concrétisation des RIA

Test de HTML5



- Pour voir si votre navigateur supporte HTML5 et savoir quelles élément/attributs sont supportés :

http://html5test.com/

- Pour une description détaillée des balises de HTML5:

http://www.w3schools.com/html5/default.asp

HTML5 support via code



- Pour voir si la navigateur qui ouvre la page supporte un certaine balise/attribut, il y a :
  - Moderniz : an open source, MIT-licensed JavaScript library
  - Pour lui faire appelle, il suffit de rajouter l'appel du script suivant à vos pages.

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<script
src="modernizr.min.js"></script>
</head>
<body>
...</body></html>
```

HTML5 support via code…



- Lorsqu'elle s'exécute, cette libraire va créer un objet global appelé modernizr au quel est associé un ensemble de propriétés avec des valeurs booléennes.
- Si votre navigateur supporte le canvas API, Modernizr.canvasproperty sera à TRUE

```
if (Modernizr.canvas) {
// let's draw some shapes!
} else {
// no native canvas support available :(
}
```

HTML5 support via code...



- Tester la fonction Javascript qui permet de manipuler l'élément html5 en question comme :
  - getContext pour CANVAS

```
function supports_canvas() {
return !!document.createElement('canvas').getContext;
}
```

  - canPlayType pour VIDEO

```
function supports_video() {
return !!document.createElement('video').canPlayType;
}
```

Les nouvelles balises de structuration de texte





HTML5 a introduit 20 nouvelles ‹balises›:

Tag	Description
<article>	For external content, like text from a news-article, blog, forum, or any other content from an external source
<aside>	For content aside from the content it is placed in. The aside content should be related to the surrounding content
<command>	A button, or a radiobutton, or a checkbox
<details>	For describing details about a document, or parts of a document
<summary>	A caption, or summary, inside the details element
<figure>	For grouping a section of stand-alone content, could be a video
<figcaption>	The caption of the figure section
<footer>	For a footer of a document or section, could include the name of the author, the date of the document, contact information, or copyright information
<header>	For an introduction of a document or section, could include navigation

Les nouvelles balises de structuration de texte (cont.)



Tag	Description
<hgroup>	For a section of headings, using <h1> to <h6>, where the largest is the main heading of the section, and the others are sub-headings
<mark>	For text that should be highlighted
<meter>	For a measurement, used only if the maximum and minimum values are known
<nav>	For a section of navigation
<progress>	The state of a work in progress
<ruby>	For ruby annotation (Chinese notes or characters)
<rt>	For explanation of the ruby annotation
<rp>	What to show browsers that do not support the ruby element
<section>	For a section in a document. Such as chapters, headers, footers, or any other sections of the document
<time>	For defining a time or a date, or both
<wbr>	Word break. For defining a line-break opportunity.

Projet DEV PORTAIL WEB



- Développer un portail Web sur un thème de votre choix qui fait intervenir :
  - Les différentes technologies abordées dans le cours :
  - HTML, CSS, Javascript, PHP, AJAX, XML, un SGBD de votre Choix.
  - Les APIs de HTML5 (CANVAS, WEB STORAGE, Web workers, etc..)
  - Les RSS FEED/ATOM
- Transformer votre application en un MASHUP en faisant intervenir au moins deux APIs disponibles sur : http://www.programmableweb.com/
- Déployer